```
# Importing all the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# loading the given dataset
df = pd.read_csv('delhivery_data.csv')

df
```

```
            data          trip_creation_time  \
0       training  2018-09-20 02:35:36.476840
1       training  2018-09-20 02:35:36.476840
2       training  2018-09-20 02:35:36.476840
3       training  2018-09-20 02:35:36.476840
4       training  2018-09-20 02:35:36.476840
...          ...                         ...
144862  training  2018-09-20 16:24:28.436231
144863  training  2018-09-20 16:24:28.436231
144864  training  2018-09-20 16:24:28.436231
144865  training  2018-09-20 16:24:28.436231
144866  training  2018-09-20 16:24:28.436231

                                    route_schedule_uuid
route_type  \
0       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting

1       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting

2       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting

3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting

4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting

...                                                    ...        ...

144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting


                      trip_uuid source_center
source_name  \
```

```
0       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC
(Gujarat)
1       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC
(Gujarat)
2       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC
(Gujarat)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC
(Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC
(Gujarat)
...                          ...           ...
...
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)

       destination_center               destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
...                   ...                           ...
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                    od_start_time  ...            cutoff_timestamp  \
0       2018-09-20 03:21:32.418600  ...         2018-09-20 04:27:55
1       2018-09-20 03:21:32.418600  ...         2018-09-20 04:17:55
2       2018-09-20 03:21:32.418600  ...  2018-09-20 04:01:19.505586
3       2018-09-20 03:21:32.418600  ...         2018-09-20 03:39:57
4       2018-09-20 03:21:32.418600  ...         2018-09-20 03:33:55
...                            ...  ...                         ...
144862  2018-09-20 16:24:28.436231  ...         2018-09-20 21:57:20
144863  2018-09-20 16:24:28.436231  ...         2018-09-20 21:31:18
144864  2018-09-20 16:24:28.436231  ...         2018-09-20 21:11:18
144865  2018-09-20 16:24:28.436231  ...         2018-09-20 20:53:19
144866  2018-09-20 16:24:28.436231  ...  2018-09-20 16:24:28.436231

        actual_distance_to_destination  actual_time  osrm_time
```

```
osrm_distance  \
0                              10.435660         14.0       11.0
11.9653
1                              18.936842         24.0       20.0
21.7243
2                              27.637279         40.0       28.0
32.5395
3                              36.118028         62.0       40.0
45.5620
4                              39.386040         68.0       44.0
54.2181
...                                  ...          ...        ...
...
144862                         45.258278         94.0       60.0
67.9280
144863                         54.092531        120.0       76.0
85.6829
144864                         66.163591        140.0       88.0
97.0933
144865                         73.680667        158.0       98.0
111.2709
144866                         70.039010        426.0       95.0
88.7319

          factor  segment_actual_time  segment_osrm_time  \
0       1.272727                 14.0               11.0
1       1.200000                 10.0                9.0
2       1.428571                 16.0                7.0
3       1.550000                 21.0               12.0
4       1.545455                  6.0                5.0
...          ...                  ...                ...
144862  1.566667                 12.0               12.0
144863  1.578947                 26.0               21.0
144864  1.590909                 20.0               34.0
144865  1.612245                 17.0               27.0
144866  4.484211                268.0                9.0

        segment_osrm_distance  segment_factor
0                     11.9653        1.272727
1                      9.7590        1.111111
2                     10.8152        2.285714
3                     13.0224        1.750000
4                      3.9153        1.200000
...                       ...             ...
144862                 8.1858        1.000000
144863                17.3725        1.238095
144864                20.7053        0.588235
144865                18.8885        0.629630
144866                 8.8088       29.777778
```

```
[144867 rows x 24 columns]
```

Problem Statement:- 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. 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

```
# shape of the dataset
df.shape
```

```
(144867, 24)
```

```
# Information of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144867 entries, 0 to 144866
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
```

```
 0   data                                144867 non-null  object
 1   trip_creation_time                  144867 non-null  object
 2   route_schedule_uuid                 144867 non-null  object
 3   route_type                          144867 non-null  object
 4   trip_uuid                           144867 non-null  object
 5   source_center                       144867 non-null  object
 6   source_name                         144574 non-null  object
 7   destination_center                  144867 non-null  object
 8   destination_name                    144606 non-null  object
 9   od_start_time                       144867 non-null  object
 10  od_end_time                         144867 non-null  object
 11  start_scan_to_end_scan              144867 non-null  float64
 12  is_cutoff                           144867 non-null  bool
 13  cutoff_factor                       144867 non-null  int64
 14  cutoff_timestamp                    144867 non-null  object
 15  actual_distance_to_destination  144867 non-null  float64
 16  actual_time                         144867 non-null  float64
 17  osrm_time                           144867 non-null  float64
 18  osrm_distance                       144867 non-null  float64
 19  factor                              144867 non-null  float64
 20  segment_actual_time                 144867 non-null  float64
 21  segment_osrm_time                   144867 non-null  float64
 22  segment_osrm_distance               144867 non-null  float64
 23  segment_factor                      144867 non-null  float64
dtypes: bool(1), float64(10), int64(1), object(12)
memory usage: 25.6+ MB
```

```
df.describe()
```

```
       start_scan_to_end_scan  cutoff_factor
actual_distance_to_destination  \
count           144867.000000  144867.000000
144867.000000
mean               961.262986     232.926567
234.073372
std               1037.012769     344.755577
344.990009
min                 20.000000       9.000000
9.000045
25%                161.000000      22.000000
23.355874
50%                449.000000      66.000000
66.126571
75%               1634.000000     286.000000
286.708875
max               7898.000000    1927.000000
1927.447705

         actual_time      osrm_time  osrm_distance         factor  \
count  144867.000000  144867.000000  144867.000000  144867.000000
```

```
mean      416.927527     213.868272     284.771297       2.120107
std       598.103621     308.011085     421.119294       1.715421
min         9.000000       6.000000       9.008200       0.144000
25%        51.000000      27.000000      29.914700       1.604264
50%       132.000000      64.000000      78.525800       1.857143
75%       513.000000     257.000000     343.193250       2.213483
max      4532.000000    1686.000000    2326.199100      77.387097

       segment_actual_time  segment_osrm_time
segment_osrm_distance  \
count        144867.000000      144867.000000           144867.00000

mean             36.196111          18.507548               22.82902

std              53.571158          14.775960               17.86066

min            -244.000000           0.000000                0.00000

25%              20.000000          11.000000               12.07010

50%              29.000000          17.000000               23.51300

75%              40.000000          22.000000               27.81325

max            3051.000000        1611.000000             2191.40370


       segment_factor
count   144867.000000
mean         2.218368
std          4.847530
min        -23.444444
25%          1.347826
50%          1.684211
75%          2.250000
max        574.250000

df.head()

       data          trip_creation_time  \
0  training  2018-09-20 02:35:36.476840
1  training  2018-09-20 02:35:36.476840
2  training  2018-09-20 02:35:36.476840
3  training  2018-09-20 02:35:36.476840
4  training  2018-09-20 02:35:36.476840

                                 route_schedule_uuid route_type  \
0  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting
1  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting
2  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting
```

```
4  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...    Carting

                  trip_uuid source_center
source_name  \
0  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)

1  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)

2  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)

3  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)

4  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)


  destination_center               destination_name  \
0       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)

                 od_start_time  ...            cutoff_timestamp  \
0  2018-09-20 03:21:32.418600  ...         2018-09-20 04:27:55
1  2018-09-20 03:21:32.418600  ...         2018-09-20 04:17:55
2  2018-09-20 03:21:32.418600  ...  2018-09-20 04:01:19.505586
3  2018-09-20 03:21:32.418600  ...         2018-09-20 03:39:57
4  2018-09-20 03:21:32.418600  ...         2018-09-20 03:33:55

   actual_distance_to_destination  actual_time  osrm_time
osrm_distance  \
0                       10.435660         14.0       11.0
11.9653
1                       18.936842         24.0       20.0
21.7243
2                       27.637279         40.0       28.0
32.5395
3                       36.118028         62.0       40.0
45.5620
4                       39.386040         68.0       44.0
54.2181

     factor  segment_actual_time  segment_osrm_time
segment_osrm_distance  \
0  1.272727                 14.0               11.0
11.9653
1  1.200000                 10.0                9.0
9.7590
2  1.428571                 16.0                7.0
10.8152
```

```
3  1.550000                  21.0                12.0
13.0224
4  1.545455                   6.0                 5.0
3.9153

   segment_factor
0        1.272727
1        1.111111
2        2.285714
3        1.750000
4        1.200000

[5 rows x 24 columns]
```

```
df.tail()
```

```
            data          trip_creation_time  \
144862  training  2018-09-20 16:24:28.436231
144863  training  2018-09-20 16:24:28.436231
144864  training  2018-09-20 16:24:28.436231
144865  training  2018-09-20 16:24:28.436231
144866  training  2018-09-20 16:24:28.436231

                                     route_schedule_uuid
route_type  \
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting

144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...    Carting


                      trip_uuid source_center
source_name  \
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H
(Haryana)

       destination_center               destination_name  \
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
```

```
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                     od_start_time  ...            cutoff_timestamp  \
144862  2018-09-20 16:24:28.436231  ...         2018-09-20 21:57:20
144863  2018-09-20 16:24:28.436231  ...         2018-09-20 21:31:18
144864  2018-09-20 16:24:28.436231  ...         2018-09-20 21:11:18
144865  2018-09-20 16:24:28.436231  ...         2018-09-20 20:53:19
144866  2018-09-20 16:24:28.436231  ...  2018-09-20 16:24:28.436231

        actual_distance_to_destination  actual_time  osrm_time
osrm_distance  \
144862                       45.258278         94.0       60.0
67.9280
144863                       54.092531        120.0       76.0
85.6829
144864                       66.163591        140.0       88.0
97.0933
144865                       73.680667        158.0       98.0
111.2709
144866                       70.039010        426.0       95.0
88.7319

          factor  segment_actual_time  segment_osrm_time  \
144862  1.566667                 12.0               12.0
144863  1.578947                 26.0               21.0
144864  1.590909                 20.0               34.0
144865  1.612245                 17.0               27.0
144866  4.484211                268.0                9.0

        segment_osrm_distance  segment_factor
144862                 8.1858        1.000000
144863                17.3725        1.238095
144864                20.7053        0.588235
144865                18.8885        0.629630
144866                 8.8088       29.777778

[5 rows x 24 columns]
```

```
# columns of the dataset
df.columns
```

```
Index(['data', 'trip_creation_time', 'route_schedule_uuid',
'route_type',
       'trip_uuid', 'source_center', 'source_name',
'destination_center',
       'destination_name', 'od_start_time', 'od_end_time',
       'start_scan_to_end_scan', 'is_cutoff', 'cutoff_factor',
```

```
       'cutoff_timestamp', 'actual_distance_to_destination',
'actual_time',
       'osrm_time', 'osrm_distance', 'factor', 'segment_actual_time',
       'segment_osrm_time', 'segment_osrm_distance',
'segment_factor'],
      dtype='object')
```

```
# unique entries present in each column
df.nunique()
```

```
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
dtype: int64
```

```
# Null values in the dataset
df.isna().sum()
```

```
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
dtype: int64
```

```
df.describe(include=object)
```

```
            data          trip_creation_time  \
count     144867                      144867
unique         2                       14817
top     training  2018-09-28 05:23:15.359220
freq      104858                         101

                                    route_schedule_uuid
route_type  \
count                                             144867     144867

unique                                              1504          2

top     thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069f...        FTL

freq                                                1812      99660


                       trip_uuid source_center
source_name  \
count                     144867        144867
144574
unique                     14817          1508
1498
top     trip-153811219535896559  IND000000ACB  Gurgaon_Bilaspur_HB
(Haryana)
freq                         101         23347
23347

       destination_center               destination_name  \
count              144867                         144606
unique               1481                           1468
top           IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
freq                15192                          15192
```

```
                      od_start_time                   od_end_time  \
count                        144867                        144867
unique                        26369                         26369
top     2018-09-21 18:37:09.322207  2018-09-24 09:59:15.691618
freq                             81                            81

           cutoff_timestamp
count                144867
unique                93180
top     2018-09-24 05:19:20
freq                     40
```

```
# converting data types of the coloumn to category and datetime for
further analysis
df['data'] = df['data'].astype('category')
df['route_type']= df['route_type'].astype('category')
df['trip_creation_time'] = pd.to_datetime(df['trip_creation_time'])
df['od_start_time'] = pd.to_datetime(df['od_start_time'])
df['od_end_time'] = pd.to_datetime(df['od_end_time'])

# Filling null values with the mode value
df['source_name'].fillna(df['source_name'].mode()[0], inplace=True)
df['destination_name'].fillna(df['destination_name'].mode()[0],
inplace=True)

# Merging of rows and aggregation of fields:-

# Since delivery details of one package are divided into several rows.
we should treat their fields after combining these rows.

grouped_df = df.groupby(by = ['trip_uuid', 'source_center',
'destination_center'],as_index = False).agg({
    'data': 'first',
    'route_type': 'first',
    'trip_creation_time': 'first',
    'source_name':'first',
    'destination_name':'last',
    'od_start_time' : 'first',
    'od_end_time' : 'first',
    'start_scan_to_end_scan': 'first',
    'actual_distance_to_destination': 'last',
    'actual_time': 'last',
    'osrm_time': 'last',
    'osrm_distance': 'last',
    'segment_actual_time': 'sum',
    'segment_osrm_time': 'sum',
    'segment_osrm_distance': 'sum'
}).reset_index()
```

```
#Calculating the time taken between od_start_time and od_end_time and
keep it as a feature and dropping the original columns, if required.
grouped_df['od_start_time'] =
pd.to_datetime(grouped_df['od_start_time'])
grouped_df['od_end_time'] = pd.to_datetime(grouped_df['od_end_time'])

grouped_df['od_total_time'] = grouped_df['od_end_time'] -
grouped_df['od_start_time']
grouped_df.drop(columns = ['od_end_time', 'od_start_time'], inplace =
True)
grouped_df['od_total_time'] = grouped_df['od_total_time'].apply(lambda
x : round(x.total_seconds() / 60.0, 2))
grouped_df['od_total_time'].head()
```

```
0    1260.60
1     999.51
2      58.83
3     122.78
4     834.64
Name: od_total_time, dtype: float64
```

```
final_grouped_df = grouped_df.groupby(by ='trip_uuid',as_index =
False).agg({'source_center' : 'first',

'destination_center' : 'last',
                                                                'data' :
'first',

'route_type' : 'first',

'trip_creation_time' : 'first',

'source_name' : 'first',

'destination_name' : 'last',

'od_total_time' : 'sum',

'start_scan_to_end_scan' : 'sum',

'actual_distance_to_destination' : 'sum',

'actual_time' : 'sum',
                                                                'osrm_time'
: 'sum',

'osrm_distance' : 'sum',

'segment_actual_time' : 'sum',
```

```
'segment_osrm_time' : 'sum',

'segment_osrm_distance' : 'sum'})

final_grouped_df.head()
```

```
                  trip_uuid source_center destination_center      data
\
0  trip-153671041653548748  IND209304AAA       IND209304AAA  training

1  trip-153671042288605164  IND561203AAB       IND561203AAB  training

2  trip-153671043369099517  IND000000ACB       IND000000ACB  training

3  trip-153671046011330457  IND400072AAB       IND401104AAA  training

4  trip-153671052974046625  IND583101AAA       IND583119AAA  training


  route_type          trip_creation_time
source_name  \
0        FTL  2018-09-12 00:00:16.535741  Kanpur_Central_H_6 (Uttar
Pradesh)
1    Carting  2018-09-12 00:00:22.886430   Doddablpur_ChikaDPP_D
(Karnataka)
2        FTL  2018-09-12 00:00:33.691250       Gurgaon_Bilaspur_HB
(Haryana)
3    Carting  2018-09-12 00:01:00.113710            Mumbai Hub
(Maharashtra)
4        FTL  2018-09-12 00:02:09.740725              Bellary_Dc
(Karnataka)

                     destination_name  od_total_time
start_scan_to_end_scan  \
0  Kanpur_Central_H_6 (Uttar Pradesh)        2260.11
2259.0
1   Doddablpur_ChikaDPP_D (Karnataka)         181.61
180.0
2       Gurgaon_Bilaspur_HB (Haryana)        3934.36
3933.0
3     Mumbai_MiraRd_IP (Maharashtra)         100.49
100.0
4       Sandur_WrdN1DPP_D (Karnataka)         718.34
717.0

   actual_distance_to_destination  actual_time  osrm_time
osrm_distance  \
0                      824.732854       1562.0      717.0
991.3523
1                       73.186911        143.0       68.0
```

```
85.1110
2                     1927.404273       3347.0     1740.0
2354.0665
3                       17.175274         59.0       15.0
19.6800
4                      127.448500        341.0      117.0
146.7918

   segment_actual_time  segment_osrm_time  segment_osrm_distance
0               1548.0             1008.0              1320.4733
1                141.0               65.0                84.1894
2               3308.0             1941.0              2545.2678
3                 59.0               16.0                19.8766
4                340.0              115.0               146.7919

# Building some features to prepare the data for actual analysis.

# Source Name:-Split and extract features out of destination. City-
place-code (State)

def extract_state(x):
    if x is None:
        return None
    l = x.split('(')
    if len(l) == 1:
        return l[0].strip()
    else:
        return l[-1].replace(')', "").strip()

final_grouped_df['source_state'] =
final_grouped_df['source_name'].apply(extract_state)
final_grouped_df['source_state'].unique()

array(['Uttar Pradesh', 'Karnataka', 'Haryana', 'Maharashtra',
       'Tamil Nadu', 'Gujarat', 'Delhi', 'Telangana', 'Rajasthan',
       'Assam', 'Madhya Pradesh', 'West Bengal', 'Andhra Pradesh',
       'Punjab', 'Chandigarh', 'Goa', 'Jharkhand', 'Pondicherry',
       'Orissa', 'Uttarakhand', 'Himachal Pradesh', 'Kerala',
       'Arunachal Pradesh', 'Bihar', 'Chhattisgarh',
       'Dadra and Nagar Haveli', 'Jammu & Kashmir', 'Mizoram',
'Nagaland',
       None], dtype=object)

def extract_city(x):
    if x is None:
        return None
    else:
        l = x.split()[0].split('_')
        if 'CCU' in x:
            return 'Kolkata'
```

```
        elif 'MAA' in x.upper():
            return 'Chennai'
        elif ('HBR' in x.upper()) or ('BLR' in x.upper()):
            return 'Bengaluru'
        elif 'FBD' in x.upper():
            return 'Faridabad'
        elif 'BOM' in x.upper():
            return 'Mumbai'
        elif 'DEL' in x.upper():
            return 'Delhi'
        elif 'OK' in x.upper():
            return 'Delhi'
        elif 'GZB' in x.upper():
            return 'Ghaziabad'
        elif 'GGN' in x.upper():
            return 'Gurgaon'
        elif 'AMD' in x.upper():
            return 'Ahmedabad'
        elif 'CJB' in x.upper():
            return 'Coimbatore'
        elif 'HYD' in x.upper():
            return 'Hyderabad'
        return l[0]

final_grouped_df['source_city'] =
final_grouped_df['source_name'].apply(extract_city)
final_grouped_df['source_city'].unique()[:100]
```

```
array(['Kanpur', 'Doddablpur', 'Gurgaon', 'Mumbai', 'Bellary',
'Chennai',
       'Bengaluru', 'Surat', 'Delhi', 'Pune', 'Faridabad', 'Shirala',
       'Hyderabad', 'Thirumalagiri', 'Gulbarga', 'Jaipur',
'Allahabad',
       'Guwahati', 'Narsinghpur', 'Shrirampur', 'Madakasira',
'Sonari',
       'Dindigul', 'Jalandhar', 'Chandigarh', 'Deoli', 'Pandharpur',
       'Kolkata', 'Bhandara', 'Kurnool', 'Bhiwandi', 'Bhatinda',
       'RoopNagar', 'Bantwal', 'Lalru', 'Kadi', 'Shahdol',
'Gangakher',
       'Durgapur', 'Vapi', 'Jamjodhpur', 'Jetpur', 'Mehsana',
'Jabalpur',
       'Junagadh', 'Gundlupet', 'Mysore', 'Goa', 'Bhopal', 'Sonipat',
       'Himmatnagar', 'Jamshedpur', 'Pondicherry', 'Anand', 'Udgir',
       'Nadiad', 'Villupuram', 'Purulia', 'Bhubaneshwar', 'Bamangola',
       'Tiruppattur', 'Kotdwara', 'Medak', 'Bangalore', 'Dhrangadhra',
       'Hospet', 'Ghumarwin', 'Agra', 'Sitapur', 'Canacona',
'Bilimora',
       'SultnBthry', 'Lucknow', 'Vellore', 'Bhuj', 'Dinhata',
       'Margherita', 'Boisar', 'Vizag', 'Tezpur', 'Koduru',
'Tirupati',
```

```
       'Pen', 'Ahmedabad', 'Faizabad', 'Gandhinagar', 'Anantapur',
       'Betul', 'Panskura', 'Rasipurm', 'Sankari', 'Jorhat', 'PNQ',
       'Srikakulam', 'Dehradun', 'Jassur', 'Sawantwadi', 'Shajapur',
       'Ludhiana', 'GreaterThane'], dtype=object)
```

```
def extract_place(x):
    if x is None:
        return None
    elif 'HBR' in x:
        return 'HBR Layout PC'
    else:
        l = x.split()[0].split('_', 1)
        if len(l) == 1:
            return 'unknown_place'
        else:
            return l[1]

final_grouped_df['source_place'] =
final_grouped_df['source_name'].apply(extract_place)
final_grouped_df['source_place'].unique()[:100]
```

```
array(['Central_H_6', 'ChikaDPP_D', 'Bilaspur_HB', 'unknown_place',
'Dc',
       'Poonamallee', 'Chrompet_DPC', 'HBR Layout PC', 'Central_D_12',
       'Lajpat_IP', 'North_D_3', 'Balabhgarh_DPC', 'Central_DPP_3',
       'Shamshbd_H', 'Xroad_D', 'Nehrugnj_I', 'Central_I_7',
       'Central_H_1', 'Nangli_IP', 'North', 'KndliDPP_D',
'Central_D_9',
       'DavkharRd_D', 'Bandel_D', 'RTCStand_D', 'Central_DPP_1',
       'KGAirprt_HB', 'North_D_2', 'Central_D_1', 'DC', 'Mthurard_L',
       'Mullanpr_DC', 'Central_DPP_2', 'RajCmplx_D', 'Beliaghata_DPC',
       'RjnaiDPP_D', 'AbbasNgr_I', 'Mankoli_HB', 'DPC', 'Airport_H',
       'Hub', 'Gateway_HB', 'Tathawde_H', 'ChotiHvl_DC', 'Trmltmpl_D',
       'OnkarDPP_D', 'Mehmdpur_H', 'KaranNGR_D', 'Sohagpur_D',
       'Chrompet_L', 'Busstand_D', 'Central_I_1', 'IndEstat_I',
'Court_D',
       'Panchot_IP', 'Adhartal_IP', 'DumDum_DPC', 'Bomsndra_HB',
       'Swamylyt_D', 'Yadvgiri_IP', 'Old', 'Kundli_H', 'Central_I_3',
       'Vasanthm_I', 'Poonamallee_HB', 'VUNagar_DC', 'NlgaonRd_D',
       'Bnnrghta_L', 'Thirumtr_IP', 'GariDPP_D', 'Jogshwri_I',
       'KoilStrt_D', 'CotnGren_M', 'Nzbadrd_D', 'Dwaraka_D',
'Nelmngla_H',
       'NvygRDPP_D', 'Gndhichk_D', 'Central_D_3', 'Chowk_D',
'CharRsta_D',
       'Kollgpra_D', 'Peenya_IP', 'GndhiNgr_IP', 'Sanpada_I',
       'WrdN4DPP_D', 'Sakinaka_RP', 'CivilHPL_D', 'OstwlEmp_D',
       'Gajuwaka', 'Mhbhirab_D', 'MGRoad_D', 'Balajicly_I',
'BljiMrkt_D',
       'Dankuni_HB', 'Trnsport_H', 'Rakhial', 'Memnagar', 'East_I_21',
       'Mithakal_D'], dtype=object)
```

```
# Destination Name: Split and extract features out of destination.
City-place-code (State)
final_grouped_df['destination_state'] =
final_grouped_df['destination_name'].apply(extract_state)
final_grouped_df['destination_state'].unique()
```

```
array(['Uttar Pradesh', 'Karnataka', 'Haryana', 'Maharashtra',
       'Tamil Nadu', 'Gujarat', 'Delhi', 'Telangana', 'Rajasthan',
       'Madhya Pradesh', 'Assam', 'West Bengal', 'Andhra Pradesh',
       'Punjab', 'Chandigarh', 'Dadra and Nagar Haveli', 'Orissa',
       'Bihar', 'Jharkhand', 'Goa', 'Uttarakhand', 'Himachal Pradesh',
       'Kerala', 'Arunachal Pradesh', 'Mizoram', 'Chhattisgarh',
       'Jammu & Kashmir', 'Nagaland', 'Meghalaya', 'Tripura', None,
       'Daman & Diu'], dtype=object)
```

```
final_grouped_df['destination_city'] =
final_grouped_df['destination_name'].apply(extract_city)
final_grouped_df['destination_city'].unique()
```

```
array(['Kanpur', 'Doddablpur', 'Gurgaon', 'Mumbai', 'Sandur',
'Chennai',
       'Bengaluru', 'Surat', 'Delhi', 'PNQ', 'Faridabad', 'Ratnagiri',
       'Bangalore', 'Hyderabad', 'Aland', 'Jaipur', 'Satna',
'Guwahati',
       'Bareli', 'Nashik', 'Hooghly', 'Sivasagar', 'Palani',
'Jalandhar',
       'Chandigarh', 'Yavatmal', 'Sangola', 'Kolkata', 'Savner',
       'Kurnool', 'Bhatinda', 'Bhiwandi', 'Barnala', 'Murbad',
'Kadaba',
       'Gulbarga', 'Naraingarh', 'Ludhiana', 'Kadi', 'Jabalpur',
       'Gangakher', 'Bankura', 'Silvassa', 'Porbandar', 'Jetpur',
       'Khammam', 'Mehsana', 'Katni', 'Una', 'Malavalli', 'HDKote',
       'Radhanpur', 'Visakhapatnam', 'Pune', 'Bhopal', 'Bhubaneshwar',
       'Allahabad', 'Sonipat', 'Himmatnagar', 'Sasaram', 'Ranchi',
       'Thiruvarur', 'Ghaziabad', 'Anand', 'Nanded', 'Noida',
'Nadiad',
       'Virudhchlm', 'Durgapur', 'Bhadrak', 'Goa', 'Balurghat',
'Hisar',
       'Tiruppattur', 'Kotdwara', 'Yellareddy', 'Halvad', 'Hospet',
       'JognderNgr', 'Kirauli', 'Dhaurahara', 'Canacona', 'Vansda',
       'Mananthavady', 'Lucknow', 'Silchar', 'Bhuj', 'Pundibari',
       'LowerParel', 'Changlang', 'Boisar', 'Tezpur', 'Koduru',
'Gudur',
       'Pen', 'Ahmedabad', 'Akbarpur', 'Purnia', 'Aurangabad',
       'Anantapur', 'Kolhapur', 'Sausar', 'Haldia', 'Dindigul',
       'Namakkal', 'Erode', 'Parvathipuram', 'Srikakulam',
'Nalasopara',
       'Pathankot', 'Malda', 'Malvan', 'Shajapur', 'Ambabadi',
'Amritsar',
       'Coimbatore', 'Jasai', 'Villupuram', 'Mettur', 'Palwal',
```

```
       'Darjeeling', 'Tiruchi', 'Dadri', 'Gotan', 'Amroha', 'Datia',
       'Dhanbad', 'Guna', 'Burhanpur', 'Mangalore', 'Margherita',
       'Chamoli', 'Ajmer', 'Pasighat', 'Mirzapur', 'Ghazipur',
'Hubli',
       'Bagalkot', 'Robertsganj', 'Haveri', 'Alwar', 'Udaipur',
       'Gandhidham', 'Solapur', 'Belgaum', 'Moga', 'Kendrpara',
'Barshi',
       'Addanki', 'Ongole', 'Sagara', 'Deoband', 'Chhatarpur',
'Siwan',
       'Rajgir', 'Thrissur', 'Mandya', 'Rishikesh', 'Manjeshwar',
       'Jamshedpur', 'Bakhtiarpur', 'Dahod', 'Tirupur', 'Karanjia',
       'Neemrana', 'Ganga', 'Arwal', 'Bhiwani', 'Kolasib',
'Midnapore',
       'Sillod', 'Nellore', 'Baharampur', 'Rawatsar', 'Kaithal',
       'Kaikaluru', 'Machilipatnam', 'Nazirpur', 'Kalwakurthy',
       'Puranpur', 'Jorhat', 'Mandi', 'Rajamundry', 'Chitradurga',
       'Draksharamam', 'Muzaffrpur', 'Akola', 'Islampur', 'Madhepura',
       'Simrahi', 'Srisailam', 'Bngnpalle', 'Tiptur', 'Bijapur',
       'Patiala', 'Bijainagar', 'Channaraya', 'Katihar', 'Ratia',
       'Makrana', 'Raigarh', 'Almora', 'Godda', 'Bayana',
'Kushinagar',
       'Dhaka', 'Kawardha', 'Bahadurgarh', 'Dhampur', 'Gorakhpur',
       'Warangal', 'Sambhal', 'Ratlam', 'Rudrapur', 'Sahatwar',
       'Balaghat', 'Raxaul', 'Narayankhed', 'Kalyandurg', 'Samana',
       'Shamli', 'Gangapur', 'Pilani', 'Dwarka', 'Kakdwip', 'Ambah',
       'Attingal', 'Surendranagar', 'Buxar', 'Anupshahar',
'Kallikkad',
       'Auraiya', 'Bhagalpur', 'Panaji', 'Raikot', 'Hapur',
'Samastipur',
       'Kaman', 'DhrmpuriTS', 'Mancherial', 'Haripad', 'Mundakayam',
       'Kollam', 'Shahada', 'AurngbadBR', 'Kanti', 'Chamorshi',
       'Pandharpur', 'Zirakpur', 'Unnao', 'Aluva', 'Kannad', 'Latur',
       'TalwandiSabo', 'Ghatampur', 'Banda', 'Konch', 'Mussoorie',
       'Tarkeshwar', 'Kalpetta', 'Phalodi', 'Tekkali', 'Sidhi',
       'Bilimora', 'Dinhata', 'Jalgaon', 'Vadodara', 'Dhubri',
'Dhule',
       'Sholinghur', 'Rajgurunagar', 'Hassan', 'Karnaprayag', 'Tangi',
       'Sirsi', 'Bailhongal', 'Sikar', 'Gonda', 'Madurai', 'Banswara',
       'Ghosi', 'Paota', 'Guruvayoor', 'Attur', 'Polur', 'Loharu',
       'Ankola', 'Karkala', 'Hanumangarh', 'Tumkur', 'Kendujhar',
       'Alappuzha', 'Kuthuparamba', 'Thirukkatupli', 'Gudalur',
       'Devarakonda', 'Ponnamaravathi', 'Karimganj', 'Khed', 'Lalpet',
       'Kalka', 'Saharsa', 'Pupri', 'Rohtak', 'Rajpalayam', 'Bina',
       'Ramanathapura', 'Meerut', 'Amalapuram', 'Bettiah',
'Chintamani',
       'Bethamangala', 'Pollachi', 'Jagdishpur', 'Sikandarpur',
       'Motihari', 'Dharapuram', 'Dinara', 'Nawalgarh', 'Champa',
'Bansi',
       'Arakkonam', 'Hoskote', 'Nedumangad', 'Rayaparthi',
```

```
'Tirunelveli',
       'Amreli', 'Tirchchndr', 'Kusumnchi', 'Deoghar', 'Jamtara',
       'Bhupalpally', 'Husnabad', 'Narsinghpur', 'Ramagundam',
'Aligarh',
       'Gwalior', 'Sakri', 'Haldwani', 'Chabua', 'Thiruvadanai',
'Manmad',
       'Siruguppa', 'Central', 'Mahasamund', 'Aonla', 'Salem',
       'Bamangola', 'Moradabad', 'CoochBehar', 'Bhalukpong', 'Jammu',
       'Medchal', 'Perundurai', 'Marakkanam', 'Bhusawal', 'Vapi',
       'Berhampur', 'Balasore', 'Didwana', 'Jagatsghpr', 'Bantwal',
       'Achrol', 'Gopalganj', 'Vadakkencherry', 'Edappal', 'Jhabua',
       'Trivandrum', 'Rampur', 'Pali', 'Shirur', 'Jalna', 'Jeypore',
       'JoguGadwal', 'Paramakudi', 'Badnaur', 'Patancheru', 'Merta',
       'Benipur', 'Jangipur', 'Shegaon', 'Fatehabad', 'Supaul',
       'Manjhaul', 'Sakleshpur', 'Sathyamangalam', 'Ooty',
       'HazratJandaha', 'Machhiwara', 'Kaptanganj', 'Davangere',
       'Lonavala', 'Baraut', 'NeemKaThana', 'DehriSone', 'Bhind',
       'Sathupally', 'Malegaon', 'Madhupur', 'Bhavnagar',
'Shindkheda',
       'Sangareddy', 'Phulera', 'Chhaygaon', 'Kopargaon', 'Raipur',
       'Asifabad', 'Chinnur', 'Bishnupur', 'Basti', 'Nakodar',
'Mansa',
       'Kashipur', 'Dola', 'Kodaikanal', 'Patan', 'Thirumalagiri',
       'Lakhnadon', 'Bobbili', 'Phulpur', 'SultnBthry', 'BilaspurHP',
       'Mahad', 'Srivijaynagar', 'Ashta', 'Pachore', 'Hajo',
'Tulsipur',
       'Chopan', 'Shillong', 'Vinukonda', 'Sujangarh', 'Shimoga',
       'Muktsar', 'Molakalmuru', 'Satara', 'Joda', 'Narnaul',
'Nandigama',
       'Sidhmukh', 'Printhlmna', 'Kekri', 'Katwa', 'Nabarangpr',
       'Pithorgarh', 'Bareilly', 'Perambalur', 'Dighwara', 'Kandi',
       'Lalgola', 'Karnal', 'Badarpur', 'Bariya', 'Bharatpur',
'Jagraon',
       'Rajpura', 'Nandurbar', 'Budhana', 'Kottayam', 'Rath',
'Shahdol',
       'Karauli', 'Khurdha', 'Hura', 'Bellary', 'Gonikoppal',
       'Dhrangadhra', 'Anakapalle', 'Duliajan', 'Phagwara',
'Kamareddy',
       'Kalpakkam', 'Dohrighat', 'Dhekiajuli', 'Kanigiri', 'Ramgarh',
       'Dharuhera', 'Arrah', 'Madhubani', 'Narsingpur', 'Rehli',
       'DehraGopipur', 'Pangodu', 'Pappadahandi', 'Saraiya', 'Dumka',
       'PaliBirsighpr', 'Punalur', 'Sujanpur', 'Bagnan', 'Fatepur',
       'Rajkot', 'Bagepalli', 'Metpally', 'Mohania', 'Ratanpura',
       'Kasaragod', 'Moodbidri', 'Manvi', 'Khedbrahma', 'Bhanvad',
       'Sawantwadi', 'Jalalpur', 'Veraval', 'Pratapgarh',
'Silapathar',
       'Chandi', 'Cochin', 'Arimbur', 'Sheikhpura', 'Chalakudy',
'Tandur',
       'Kotagiri', 'Nowda', 'Vijayawada', 'Benipatti', 'Padrauna',
```

```
       'Chaksu', 'Panskura', 'Chimkurthy', 'Giridih', 'Botad',
'Udgir',
       'Junagadh', 'Tezu', 'Jaisalmer', 'Islampure', 'Agartala',
       'Mainpuri', 'Kathua', 'Chandauli', 'Dharwad', 'Aizawl',
'Uchila',
       'Tikamgarh', 'Beed', 'Koraput', 'Karad', 'Mannargudi', 'Dhone',
       'Buldhana', 'Parwanoo', 'Kandukur', 'Morgram', 'Mungeli',
'Theni',
       'PaontSahib', 'Bilaspur', 'Areacode', 'Lalru', 'Kaliyaganj',
       'Paranpur', 'Sihora', 'Shivpuri', 'Nagarcoil', 'Gondal',
'Tirpur',
       'Manuguru', 'Bhota', 'Dhrmsthala', 'Jewar', 'Tonk',
'Rghunthpur',
       'Pavagada', 'Puttur', 'Sinnar', 'Bhandara', 'Bolpur',
'Parbhani',
       'Suratgarh', 'Gundlupet', 'Bodhan', 'Chidambaram', 'Agra',
       'Gangavathi', 'Palakonda', 'Palasa', 'Kharagpur', 'Tirupati',
None,
       'Krishnagiri', 'Erandol', 'LakhimpurN', 'Gopiganj', 'Baripada',
       'Jehanabad', 'Palamaner', 'Chanapatna', 'Nohar', 'Asansol',
       'Umerkote', 'Chapra', 'Gangarmpur', 'Aranthangi', 'Shamshabad',
       'Kullu', 'Jalalabad', 'Khanna', 'Kalluvathukal', 'Samsi',
       'Wankaner', 'Oriyur', 'Vizianagaram', 'BariSadri', 'Pilibanga',
       'Vishakhapatnam', 'Lodhan', 'Moranhat', 'Barauni', 'Dholpur',
       'Ghanpur', 'Mangaldoi', 'Gahmar', 'Chiraiyakot', 'ChrkhiDdri',
       'Baddi', 'Degana', 'Sultana', 'Parakkdavu', 'Seoni',
'Anupgarh',
       'Sindagi', 'Sedam', 'Nakhatrana', 'MirzapurWB', 'Mathabhang',
       'Khatra', 'Champhai', 'Rona', 'Shahganj', 'Jowai',
'Chittaurgarh',
       'Arani', 'Jhajjar', 'Malappuram', 'Kallachi', 'Modinagar',
       'Atmakur', 'Berhampore', 'Ramnagar', 'Buhana', 'Kahalgaon',
       'Patran', 'Deoria', 'Gadchiroli', 'Neemuch', 'Deoli', 'Sonari',
       'Digboi', 'Namsai', 'Sitamau', 'Churhat', 'Dahanu', 'Khanapur',
       'Balrampur', 'Varanasi', 'Mandapeta', 'Araria', 'Nuzvid',
       'Helencha', 'Bangarapet', 'Khambhalia', 'Jagtial',
'Jammikunta',
       'Soro', 'Contai', 'Anjar', 'Howrah', 'Lakhipur', 'Chamba',
'Mau',
       'Ramnthpurm', 'Mehkar', 'SundarNgr', 'Malerkotla', 'Jadcherla',
       'Kasganj', 'Athani', 'Mahbubabad', 'Manikchak', 'Umaria',
       'Karukachal', 'Jalore', 'Koppa', 'Khanakul', 'Mandsaur',
'Ranipet',
       'Jairampur', 'Dhemaji', 'Hathras', 'Sirsa', 'Parbatsar',
       'Rajgangpur', 'Bargarh', 'Khanpur', 'Sirohi', 'Chalisgaon',
       'Dabhoi', 'RampuraPhul', 'Bhilad', 'Bhatkal', 'Betnoti',
'Raichur',
       'Chikmagalur', 'Ranikhet', 'Mathura', 'Markapur', 'Balangir',
       'Panipat', 'Dharmapuri', 'Lalitpur', 'Modasa', 'Jasdan',
```

```
'Aliganj',
       'Khalilabad', 'Nagapttinm', 'Jahu', 'Akhnoor', 'Jassur',
'Nagpur',
       'Aunrihar', 'Dehradun', 'Rayadurgam', 'Razole', 'Jhanjharpur',
       'Bongaon', 'Sumerpur', 'Valsad', 'Gangarampr', 'Gujiliam',
'Gomoh',
       'Arambag', 'SrinagarUK', 'Phusro', 'Shadnagar', 'Vadakara',
       'Firozabad', 'Sultanganj', 'Atapadi', 'RoopNagar',
'Giddarbaha',
       'Barmer', 'Chodavaram', 'Kittur', 'Vellore', 'Gohana',
'Koyilandy',
       'Tirurangadi', 'KharagpurBR', 'Kolar', 'Raver', 'Paradip',
       'Khatauli', 'Kozhenchery', 'Chandpur', 'Kattappana', 'Rajgarh',
       'Ambegaon', 'Udumalpet', 'Raiganj', 'Mothkur', 'Nirsa',
       'Venktagiri', 'Manbazar', 'Udala', 'Cuttack', 'Sonepur',
       'Faridpur', 'Dumraon', 'Kanker', 'Kakinada', 'Luxettipet',
       'Bellmpalli', 'Chanchal', 'Shirpur', 'Oddnchtram', 'Nichlaul',
       'Mysore', 'Kodad', 'Khambhat', 'Umreth', 'Tilhar', 'Chetpet',
       'Rewari', 'Cuddapah', 'Pazhayannur', 'Sundargarh', 'Baruipur',
       'Anandnagar', 'Khetri', 'Manthani', 'Thakurdwara', 'Malout',
       'Chincholi', 'Daman', 'Uthangarai', 'Gosainganj', 'Chikblapur',
       'Farrukhbad', 'Durg', 'Thachnttukra', 'Chikodi', 'Ranaghat',
       'Munger', 'Bijnor', 'Lunawada'], dtype=object)
```

```
final_grouped_df['destination_place'] =
final_grouped_df['destination_name'].apply(extract_place)
final_grouped_df['destination_place'].unique()
```

```
array(['Central_H_6', 'ChikaDPP_D', 'Bilaspur_HB', 'MiraRd_IP',
       'WrdN1DPP_D', 'Poonamallee', 'Vandalur_Dc', 'HBR Layout PC',
       'Central_D_3', 'Bhogal', 'unknown_place', 'MjgaonRd_D',
       'Nelmngla_H', 'Uppal_I', 'RazaviRd_D', 'Central_I_7',
       'Central_I_2', 'Hub', 'SourvDPP_D', 'Varachha_DC',
'TgrniaRD_I',
       'DC', 'Gokulam_D', 'Babupaty_D', 'Bomsndra_HB', 'Alwal_I',
       'RjndraRd_D', 'Mehmdpur_H', 'Sanpada_I', 'JajuDPP_D',
       'Central_DPP_2', 'Dankuni_HB', 'Wagodha_D', 'AbbasNgr_I',
       'Balabhgarh_DPC', 'DPC', 'Mankoli_HB', 'Shamshbd_H',
'SnkunDPP_D',
       'Kharar_DC', 'AnugrDPP_D', 'Nehrugnj_I', 'Ward2DPP_D',
       'MilrGanj_HB', 'KaranNGR_D', 'Adhartal_IP', 'Poonamallee_HB',
       'Busstand_D', 'BhowmDPP_D', 'Samrvrni_D', 'NSTRoad_I',
       'Panchot_IP', 'Bargawan_DC', 'KGAirprt_HB', 'Mamlatdr_DC',
       'SulthnRd_D', 'Jogeshwri_L', 'BegurRD_D', 'Santalpr_D',
       'Gajuwaka_IP', 'Tathawde_H', 'Trnsport_H', 'Central_H_1',
       'Kundli_H', 'Rohini_DPC', 'Bypasrd_D', 'Mohan_Nagar_DPC',
       'Madhavaram_L', 'Vaghasi_IP', 'Aswningr_I', 'Sec', 'SelamRd_D',
       'Central_I_1', 'Porur_DPC', 'Perungudi_DPC', 'AkhirDPP_D',
       'IndstlAr_I', 'Raiprvlg_L', 'Jhilmil_L', 'KoilStrt_D',
'Nzbadrd_D',
```

```
       'JKRoad_D', 'Mayapuri_PC', 'Hoodi_IP', 'CrossRD_D', 'Dhelu_D',
       'Central_DPP_3', 'AchneraRD_D', 'JPNagar_Pc', 'KHRoad_I',
       'TahsilRD_D', 'Kishangarh_DPC', 'CharRsta_D',
'CottonGreen_DPC',
       'CikhliRD_D', 'PunjabiB_L', 'Central_D_1', 'Kengeri_IP',
'Indira',
       'Peenya_IP', 'Sirikona_H', 'Khandeshwar_Dc', 'Alwal_L',
       'StatonRD_D', 'CP', 'OstwlEmp_D', 'Mhbhirab_D', 'MGRoad_D',
       'Bngisheb_D', 'Sector63_L', 'BljiMrkt_D', 'Bnnrghta_L',
       'Beliaghata_DPC', 'Airport_H', 'Lake', 'East', 'Memnagar',
       'Mumbra_DC', 'Satellite', 'Auliyapr_D', 'Ulhasngr_DC',
'East_H_1',
       'Pawane_L', 'Kalyan', 'Central_H_2', 'Mthurard_L', 'New',
       'KamaStrt_I', 'RPC', 'Peenya_L', 'Shivaji_I', 'Central_DPP_1',
       'Central_D_2', 'Central_D_12', 'Rkcomplx_DC', 'Mohali',
       'Chrompet_L', 'Central_D_9', 'Chrompet_DPC', 'Kuslpram_I',
       'Sixmile', 'Chandmari', 'VarunCly_DC', 'krshnPly_DC',
'BllvMarg_D',
       'Central_D_10', 'MhpraRD_D', 'NgrNigam_DC', 'Egmore_DPC',
       'Nangli_IP', 'Karayam_H', 'JNPT_D', 'Lajpat_IP', 'Thirumtr_IP',
       'Madhavaram_DPC', 'RTOroad_D', 'Hillcard_DC', 'Samyaprm_D',
       'Blbgarh_DC', 'Manesar', 'ICDCant_D', 'B_RPC', 'DKLogDPP_D',
       'SamitiRd_D', 'Kundli_P', 'TrtllaRD_L', 'TownDPP_D',
'Kalynpur_I',
       'Raghogrh_D', 'StRoad_D', 'Kuntikna_H', 'CivilHPL_D',
'CGRoad_D',
       'FoySGRRD_I', 'NwYlhnka_DC', 'MissonRd_D', 'JangiRd_D',
       'Kaithwal_D', 'Adargchi_IP', 'Jogshwri_I', 'Sector4_D',
       'ArtoDPP_D', 'GuttalRD_D', 'Mangri_I', 'Sector1A_IP', 'Dc',
       'Shamshbd_P', 'Dankuni_P', 'Kapleswr_D', 'Okhla_PC',
'Nangli_L',
       'Kurduwdi_D', 'Oilmilrd_D', 'SubhVRTL_I', 'HUB',
'Patparganj_DPC',
       'Chndivli_PC', 'KSClny_DC', 'Vardhard_D', 'Mankoli_GW',
       'Chrompet_PC', 'Bilaspur_P', 'Pandesra_Gateway', 'Ramvlg_D',
       'HnmntNgr_D', 'PnditNGR_D', 'Poothole_D', 'Tolichwk_I',
       'Central_I_3', 'BypassRD_D', 'Kapshera_L', 'RIICO_L',
'Koliplm_I',
       'Sarubali_D', 'Rcocmplx_D', 'Rozapar_D', 'Diakkawn_D',
'Talkui_D',
       'ZebaTWR_D', 'North_R_8', 'Chuanpur_I', 'ShivmDPP_D',
'Atapaka_D',
       'VidyaNGR_D', 'Mundhawa_L', 'MdothdRD_D', 'RicMilRd_D',
       'PlaceCol_D', 'AtoNgrRd_I', 'Anaipeta_D', 'Bbganj_I',
'Gaurkshn_I',
       'Shantanu_D', 'Wazirpur_L', 'Krishnpr_D', 'Bazar_D',
'Sishumdr_D',
       'Enkndla_D', 'YTRd_D', 'RKComplx_D', 'KirtiNgr_D',
'GainMrkt_L',
```

```
       'patna_D', 'Shyndco_D', 'SchdvDPP_D', 'Mwalibad_D',
'MithmdRd_D',
       'SuzkiSrv_D', 'PushPlza_D', 'LalBagh_D', 'KasyaDPP_D',
'East_I_21',
       'PchpkrRD_D', 'TrnsptNGR_D', 'NaginaRD_D', 'Matriprm_IP',
       'HunterRd_I', 'Khwssrai_D', 'Khjurwli_DC', 'North_I_4',
       'UdhamNgr_H', 'PnchmDPP_D', 'Kosmi_D', 'KairiyaT_D',
'Datatrya_D',
       'Shankrpa_D', 'PODPP_D', 'KarnalRd_D', 'Ward6DPP_D',
'StnRoad_DC',
       'KlngrDPP_D', 'MrenTirh_D', 'Tejpal_I', 'Sarswati_D',
'DcntCLY_D',
       'Mutyvila_D', 'AryaNagr_D', 'Kharghar_D', 'Pbroad_DC',
       'Swargash_D', 'Haripur_D', 'BsstdDPP_D', 'HanumDPP_D',
'Hitech_D',
       'Kumrpurm_D', 'Nandrbar_D', 'Mahindra_D', 'Khar', 'RajCmplx_D',
       'KamnHbRD_I', 'VikasRam_D', 'Peedika_H', 'KolheDPP_D',
       'Srnwsngr_D', 'Wardno3_D', 'StatinRD_D', 'Rawatpur_D',
       'HydRoad_DC', 'GayatriN_D', 'GovndNgr_DC', 'Bokule_H',
       'BrlwgDPP_D', 'Chikdply_I', 'UBamdDPP_D', 'Naraynpr_D',
'Ward19_D',
       'PalikDPP_D', 'Phaphamu_DC', 'AmvdiDPP_D', 'Padra_D',
'WrdN4DPP_D',
       'Chandkheda_Dc', 'Skynet_INT', 'Vepmpttu_DC', 'Karelibaug_DPC',
       'Paschim_DC', 'LB-Nagar_Dc', 'CotnGren_M', 'MiraRoad_M',
       'Tetultol_D', 'MIDCAvdn_I', 'ArkonmRD_D', 'Chakan_D',
'Pandrnga_I',
       'Umalodge_D', 'SriDPP_D', 'Vidygiri_D', 'NamoNagr_D',
'FatehpRd_I',
       'KotwaliN_D', 'Kappalur_H', 'KhandDPP_D', 'Jamalpur_D',
       'SmClyDPP_D', 'ManhrBld_D', 'DumDum_DPC', 'KaaduRd_D',
       'StationRD_D', 'BstndDPP_D', 'Kakrmath_D', 'MarketRd_D',
       'Veersagr_I', 'Sirjudin_D', 'Pazhvedu_D', 'IdstrlAr_D',
'Poondi_D',
       'kalmpuza_D', 'DindiRD_D', 'Puduvalvu_D', 'Alngjuri_D',
'Mahad_D',
       'PriyrNGR_D', 'Pinjore_DC', 'Gangjala_D', 'SngihiRD_D',
       'AshkTalk_D', 'Srvdycwk_D', 'Vijdurg_D', 'BypassRd_D',
       'FshryOFC_D', 'Venkatsa_DC', 'Wardno13_D', 'PlsrdDPP_D',
       'RajaBzr_D', 'Techrcly_D', 'Wardno7_D', 'NavldiDPP_D',
       'Brplicwk_D', 'GangDPP_D', 'Banshkri_DC', 'HousngBd_D',
       'Arsprmbu_D', 'Perkadrd_D', 'VdkkuSrt_I', 'Rajula_DC',
       'Shnmgprm_D', 'SKRoad_D', 'Uppal_L', 'Barmasia_D', 'D',
       'JwahrNGR_D', 'Greenmkt_D', 'KndliDPP_D', 'Pdmavati_D',
       'KhirByps_I', 'RjndrNgr_DC', 'HrihrNgr_I', 'DhuleRoad_D',
       'PiliKoti_D', 'CollgeRD_D', 'North', 'Thiruvlr_DC',
'RamnadRD_D',
       'Malegaon_D', 'South_D_12', 'Central_D_7', 'Wrd12DPP_D',
       'Ponda_Dc', 'RajpurRD_D', 'KdidmCLY_D', 'Psthrjhr_D', 'PC',
```

```
       'Khenewa_D', 'TrnptNgr_L', 'MROoffce_D', 'Trimulgherry_Dc',
       'Panvel_D', 'Viveka_DC', 'MJRDPP_D', 'Central_I_4',
'Samarth_D',
       'IndEstat_I', 'Khajuria_I', 'Ganeshwr_D', 'KatlaDPP_D',
       'Markndpr_D', 'Trmltmpl_D', 'Mehmdpur_P', 'BgwriDPP_D',
       'Bsavangr_D', 'GopalDPP_D', 'Robinson_D', 'PonaniRD_D',
'Ward11_D',
       'Mnanthla_H', 'RoshnBgh_I', 'Nayagaon_I', 'BhgyaNgr_D',
       'Kelasahi_D', 'ColctrOf_D', 'VagaiNgr_D', 'BhmrdDPP_D',
       'Ameenpur_I', 'Javahar_D', 'SChwkDPP_D', 'BhunaDPP_D',
       'SadrHsptl_D', 'Purbari_D', 'RgvdrDPP_D', 'Mlydpthr_D',
       'Davisdle_D', 'HajiprRD_D', 'MnBzrDPP_D', 'Subshngr_D',
       'NngrgnRd_D', 'SrnprHwy_D', 'Margao_Dc', 'Tejpal_M',
'War5DPP_D',
       'Dilliyan_D', 'BhrolDPP_D', 'Rawlgaon_D', 'Sitarmrd_D',
       'Kadugodi_D', 'Mahuva_DC', 'Shahdara', 'KakaCplx_D',
'Pothredy_D',
       'NarenaRD_D', 'Sriperumbudur_Dc', 'GwhRDDPP_D', 'NkshtrPz_D',
       'Barwala', 'Central_D_5', 'PaikjNGR_D', 'AsnsdhRD_D',
'StnRdDPP_D',
       'GndhiNgr_D', 'ChowkDPP_D', 'GreenVly_D', 'Vaishali_D',
       'ChainDPP_D', 'Athithnr_DC', 'Xroad_D', 'ColegRd_D',
'Shekhpur_D',
       'Kollgpra_D', 'Indsarea_D', 'Govndsgr_D', 'BhwanDPP_D',
       'ShantiNg_D', 'Shop3DPP_D', 'Sardala_D', 'MohnVRTL_D',
       'MohanNgr_C', 'Manikndm_H', 'PreetDPP_D', 'Kothapet_D',
       'ChtrGIDC_IP', 'LdnunDPP_D', 'Mhdiptnm_C', 'KnsgraRD_D',
       'Bomsndra_PC', 'Chndrlpd_D', 'MnbzrDPP_D', 'EmsPnmbi_D',
       'LxmntDPP_D', 'BSarani_D', 'PhdofDPP_D', 'Kumud_D',
'goplpurm_D',
       'SadarHPL_D', 'DohalDPP_D', 'KrsprDPP_D', 'Konapara_D',
       'BgnprDPP_D', 'DhuleRd_D', 'Mughlpra_D', 'Sohagpur_D',
       'HnsChowk_D', 'MdhsnDPP_D', 'Thomas_D', 'NvygRDPP_D',
'Kothuru_D',
       'StatonRd_D', 'Lovely_D', 'Potheri', 'Devenply_I',
'Chatrpr_DC',
       'Sadras_D', 'AzmrdDPP_D', 'Wardno5_D', 'Tiglgndi_D',
'HotelPrk_D',
       'Pandriba_L', 'Katira_D', 'Bardivan_D', 'BypRDDPP_D',
'JrjolDPP_D',
       'Ward7DPP_D', 'East_L_23', 'Wardno4_D', 'Dudhani_D',
'CtyLgDPP_D',
       'PostofJN_D', 'SainkSCL_D', 'Harop_D', 'NH117_D', 'PedakRd_P',
       'Solaiprm_D', 'TBCross_D', 'GunjRDPP_D', 'Bhabua_D',
'MubarDPP_D',
       'Todapur_DC', 'Nullipad_D', 'MrdiVlge_D', 'APMCYard_D',
       'Patelfli_D', 'Mainroad_D', 'LaxmiNgr_D', 'HSR_Layout_PC',
       'Mhimapur_D', 'Nimachrd_D', 'Hejunagr_D', 'SH78_D',
'Kdvantra_D',
```

```
       'Veluthur_D', 'Bgwtichk_D', 'SurbhiTh_D', 'SnthiNGR_D',
       'CroslySRT_D', 'AmtlaDPP_D', 'Rynapadu_H', 'WardNo1_D',
       'BawliDPPP_D', 'ModelTwn_P', 'NraynDPP_D', 'TonkRoad_D',
       'MSRClgRd_D', 'Shivalya_D', 'JatniDPP_D', 'Mangol_DC',
       'Mullanpr_DC', 'NlgaonRd_D', 'Farmnala_D', 'Gopa3PL_D',
       'ShbdnDPP_D', 'Chmpmura_I', 'Agraroad_I', 'Sholinganallur_Dc',
       'North_D_3', 'BOB_D', 'ShubsNGR_D', 'HunthrVg_I', 'MndiRoad_D',
       'Sector02_C', 'JalnaRd_D', 'GhtimDPP_D', 'Mundhe_D',
'Vadasari_D',
       'RmNyrDPP_D', 'Thsil3PL_D', 'LICOffce_D', 'Kntgorya_D',
       'Panderia_D', 'Rathnam_D', 'Gurudwar_D', 'GrmNgriya_D',
       'Puthalam_D', 'OnkarDPP_D', 'FatprDPP_D', 'Nijgan_D',
'KtnRdDPP_D',
       'AsrplmRd_DC', 'Mapusa', 'Palladam_DC', 'AskNagar_D',
'BpassDPP_D',
       'Beltngdi_D', 'SJRoad_D', 'barkarRd_D', 'PnukndRD_D',
'Darbe_DC',
       'Patel', 'KaremDPP_D', 'WebelDPP_D', 'Rjndrngr_D', 'Old',
       'Swamylyt_D', 'SuryaDPP_D', 'ARBNorth_DC', 'Aliganj',
'PhrmPlza_D',
       'VadaiDPP_D', 'Nerul_D', 'Balajicly_I', None, 'BnglorRd_D',
       'BsStdDPP_D', 'SashPhkn_D', 'KalikDPP_D', 'GodamDPP_D',
       'Madarpur_D', 'Lakshmi_D', 'NagarDPP_D', 'Whitefld_L',
       'KhdimDPP_D', 'HsptlRod_D', 'Lngrguda_D', 'AkhraBzr_D',
       'farukngr_D', 'Pariplly_D', 'RatuaDPP_D', 'Bandel_D',
'JivanDPP_D',
       'Vllyaprm_D', 'NcsRd_DC', 'BhmprDPP_D', 'GoalpDPP_D',
'TiloiDPP_D',
       'Wardno10_D', 'GtRoad_D', 'Palakrty_D', 'LNBRoad_D',
'Wardnor4_D',
       'KamalDPP_D', 'PuranDPP_D', 'ByePass_D', 'BhukrdPP_D',
'KeRoad_D',
       'Kidwai_D', 'PrmNrDPP_D', 'KalyanNg_D', 'DBRCmplx_D',
'ClgRDDPP_D',
       'VidyaDPP_D', 'Pshimpra_D', 'Sec-83_DC', 'KoralDPP_D',
       'AwmpiVng_D', 'GadagRD_D', 'KcharaRD_D', 'Ldthlabh_D',
       'KrthiKyn_D', 'Munduprm_D', 'ZamQuatr_D', 'SikriKla_DC',
       'IndraNgr_D', 'Chithbrm_D', 'BhwniGnj_D', 'CourtDPP_D',
       'NdiaTola_D', 'MheshNGR_D', 'CCRoad_D', 'KarjuDPP_D',
'JyotiNgr_D',
       'HelipadRD_D', 'KetyDPP_D', 'PBRDDPP_D', 'MahmurGj_IP',
'Mainrd_D',
       'Chtrpuza_D', 'Wardn13_D', 'Ward17_D', 'ColnyDPP_D',
'Pettah_D',
       'KolarRd_D', 'JdswarRD_D', 'East_I_20', 'Aravind_D',
'ConduDPP_D',
       'Mylapore', 'UttarDPP_D', 'Kanakpur_D', 'Salap_DC', 'Kolar',
       'Bangotu_D', 'Kadipur', 'SohnaRd_D', 'BaliaMod_D',
'TnhbBlkC_D',
```

```
       'SagarDPP_D', 'Bhogpur_D', 'Badeplly_D', 'BnkrGate_D',
       'VikrmMah_D', 'Yellanda_D', 'AgrohDPP_D', 'Enayetpr_D',
       'MnimlaRd_D', 'RoopNgr_D', 'Jharia_DC', 'Sangetha_D',
'Ramnagar_D',
       'YashDPP_D', 'East_D_8', 'MBTRd_DC', 'Mdiclcly_D', 'WardNo3_D',
       'AnprnDPP_D', 'AjmhwDPP_D', 'JJCpxDPP_D', 'NehruNGR_D',
       'Ukkadam_D', 'Nrsampt_D', 'Ricco_D', 'BhadgDPP_D',
'Poonamallee_L',
       'Umargaon_DC', 'KmkshBul_D', 'Kadtmpty_D', 'MhliaDPP_D',
       'LSRoad_DC', 'Central_L_8', 'Subhash_D', 'Kondapur_D',
'Muktsar_D',
       'Ghansoli_DC', 'Kovaipudur_Dc', 'Lajwanti', 'VidyaNgr_D',
       'Central_H_4', 'Rjndpara_D', 'RjghatRd_D', 'MotiDPP_D',
       'KaimgnjRD_D', 'MrgnjDPP_D', 'Sttyapar_D', 'Sulgwan_D',
       'ThthiCwk_D', 'AadiDPP_D', 'Jaripatk_DC', 'Sarjapur_D',
       'RailGate_D', 'Shanthi_D', 'Nagar_D', 'Sookhtal_D', '',
       'Gobindgarh_DC', 'SukntDPP_D', 'JiswlDPP_D', 'West_Dc',
       'BazarDPP_D', 'FulbaDPP_D', 'Parai_D', 'KhsmiDPP_D',
'BalibDPP_D',
       'Srikot_D', 'RhmgjDPP_D', 'Mandodi_D', 'Sudmpuri_D',
'Vidyangr_D',
       'Bhandup', 'ChotiHvl_DC', 'Nehru3PL_D', 'BaljiDPP_D',
'ColageRD_D',
       'keshod_DC', 'GndhiNgr_IP', 'DvlalDPP_D', 'Pshrikvu_D',
       'Kooriyad_D', 'Ambedkar_D', 'Truptingr_D', 'Udyabata_D',
       'TilakNgr_D', 'MunplDPP_D', 'Cherukole_D', 'NorprRD_D',
       'Palikval_D', 'SadulDPP_D', 'Manchar_D', 'Artclgrd_D',
       'SubrtDPP_D', 'ShantiDPP_D', 'TirupthiRd_D', 'Royapuram',
       'NagplDPP_D', 'SliprDPP_DC', 'BrezeDPP_D', 'ShjnprRD_D',
       'Nishangr_D', 'KrisnKunj_D', 'ShivaDPP_D', 'BasthDPP_D',
       'BargaDPP_D', 'KrantiNgr_D', 'Palani_D', 'Yadvgiri_IP',
       'MotvdDPP_D', 'Dakor_DC', 'East_D_7', 'SingCLNY_D',
'Blmrgnst_D',
       'YuktiDPP_D', 'AlathurRD_D', 'DiyoDPP_D', 'Bnsibtla_D',
       'LohiaDPP_D', 'Chaitnya_D', 'Fathuluh_D', 'RgstrOFC_D',
       'Sanpada_CP', 'Pakrela_D', 'ShntiSgr_D', 'Pnchlght_D',
'Bhilai_DC',
       'Nattukal_D', 'ArickDPP_D', 'Kaura_D', 'NaginaRd_D',
'VrdhriRD_D'],
      dtype=object)
```

```
# Trip_creation_time: Extract features like month, year and day etc...

final_grouped_df['trip_creation_time'] =
pd.to_datetime(final_grouped_df['trip_creation_time'])

final_grouped_df['trip_creation_date']=final_grouped_df['trip_creation
_time'].dt.date
final_grouped_df['trip_creation_date'].head()
```

```
0    2018-09-12
1    2018-09-12
2    2018-09-12
3    2018-09-12
4    2018-09-12
Name: trip_creation_date, dtype: object
```

```
final_grouped_df['trip_creation_day']=final_grouped_df['trip_creation_time'].dt.day
final_grouped_df['trip_creation_day'].head()
```

```
0    12
1    12
2    12
3    12
4    12
Name: trip_creation_day, dtype: int32
```

```
final_grouped_df['trip_creation_month']=final_grouped_df['trip_creation_time'].dt.month
final_grouped_df['trip_creation_month'].head()
```

```
0    9
1    9
2    9
3    9
4    9
Name: trip_creation_month, dtype: int32
```

```
final_grouped_df['trip_creation_year']=final_grouped_df['trip_creation_time'].dt.year
final_grouped_df['trip_creation_year'].head()
```

```
0    2018
1    2018
2    2018
3    2018
4    2018
Name: trip_creation_year, dtype: int32
```

```
final_grouped_df['trip_creation_hour']=final_grouped_df['trip_creation_time'].dt.hour
final_grouped_df['trip_creation_hour'].head()
```

```
0    0
1    0
2    0
3    0
4    0
Name: trip_creation_hour, dtype: int32
```

```
final_grouped_df['trip_creation_week'] =
final_grouped_df['trip_creation_time'].dt.isocalendar().week
final_grouped_df['trip_creation_week'].head()
```

```
0    37
1    37
2    37
3    37
4    37
Name: trip_creation_week, dtype: UInt32
```

```
final_grouped_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14817 entries, 0 to 14816
Data columns (total 29 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   trip_uuid                       14817 non-null  object
 1   source_center                   14817 non-null  object
 2   destination_center              14817 non-null  object
 3   data                            14817 non-null  object
 4   route_type                      14817 non-null  object
 5   trip_creation_time              14817 non-null  datetime64[ns]
 6   source_name                     14807 non-null  object
 7   destination_name                14809 non-null  object
 8   od_total_time                   14817 non-null  float64
 9   start_scan_to_end_scan          14817 non-null  float64
 10  actual_distance_to_destination  14817 non-null  float64
 11  actual_time                     14817 non-null  float64
 12  osrm_time                       14817 non-null  float64
 13  osrm_distance                   14817 non-null  float64
 14  segment_actual_time             14817 non-null  float64
 15  segment_osrm_time               14817 non-null  float64
 16  segment_osrm_distance           14817 non-null  float64
 17  source_state                    14807 non-null  object
 18  source_city                     14807 non-null  object
 19  source_place                    14807 non-null  object
 20  destination_state               14809 non-null  object
 21  destination_city                14809 non-null  object
 22  destination_place               14809 non-null  object
 23  trip_creation_date              14817 non-null  object
 24  trip_creation_day               14817 non-null  int32
 25  trip_creation_month             14817 non-null  int32
 26  trip_creation_year              14817 non-null  int32
 27  trip_creation_hour              14817 non-null  int32
 28  trip_creation_week              14817 non-null  UInt32
dtypes: UInt32(1), datetime64[ns](1), float64(9), int32(4), object(14)
memory usage: 3.0+ MB
```

```
final_grouped_df.describe().T
```

	count	mean
trip_creation_time	14817	2018-09-22 12:44:19.555167744
od_total_time	14817.0	531.69763
start_scan_to_end_scan	14817.0	530.810016
actual_distance_to_destination	14817.0	164.477838
actual_time	14817.0	357.143754
osrm_time	14817.0	161.384018
osrm_distance	14817.0	204.344689
segment_actual_time	14817.0	353.892286
segment_osrm_time	14817.0	180.949787
segment_osrm_distance	14817.0	223.201161
trip_creation_day	14817.0	18.37079
trip_creation_month	14817.0	9.120672
trip_creation_year	14817.0	2018.0
trip_creation_hour	14817.0	12.449821
trip_creation_week	14817.0	38.295944

	min
trip_creation_time	2018-09-12 00:00:16.535741
od_total_time	23.46
start_scan_to_end_scan	23.0
actual_distance_to_destination	9.002461
actual_time	9.0
osrm_time	6.0
osrm_distance	9.0729
segment_actual_time	9.0
segment_osrm_time	6.0
segment_osrm_distance	9.0729
trip_creation_day	1.0
trip_creation_month	9.0
trip_creation_year	2018.0
trip_creation_hour	0.0
trip_creation_week	37.0

```
                                                      25%  \
trip_creation_time              2018-09-17 02:51:25.129125888
od_total_time                                          149.93
start_scan_to_end_scan                                  149.0
actual_distance_to_destination                       22.837239
actual_time                                             67.0
osrm_time                                               29.0
osrm_distance                                         30.8192
segment_actual_time                                     66.0
segment_osrm_time                                       31.0
segment_osrm_distance                                 32.6545
trip_creation_day                                       14.0
trip_creation_month                                      9.0
trip_creation_year                                    2018.0
trip_creation_hour                                       4.0
trip_creation_week                                      38.0

                                                      50%  \
trip_creation_time              2018-09-22 04:02:35.066945024
od_total_time                                          280.77
start_scan_to_end_scan                                  280.0
actual_distance_to_destination                       48.474072
actual_time                                             149.0
osrm_time                                                60.0
osrm_distance                                         65.6188
segment_actual_time                                     147.0
segment_osrm_time                                        65.0
segment_osrm_distance                                 70.1544
trip_creation_day                                        19.0
trip_creation_month                                       9.0
trip_creation_year                                     2018.0
trip_creation_hour                                       14.0
trip_creation_week                                       38.0

                                                      75%  \
trip_creation_time              2018-09-27 19:37:41.898427904
od_total_time                                          638.2
start_scan_to_end_scan                                  637.0
actual_distance_to_destination                     164.583208
actual_time                                             370.0
osrm_time                                               168.0
osrm_distance                                         208.475
segment_actual_time                                     367.0
segment_osrm_time                                       185.0
segment_osrm_distance                                 218.8024
trip_creation_day                                        25.0
trip_creation_month                                       9.0
trip_creation_year                                     2018.0
trip_creation_hour                                       20.0
```

```
trip_creation_week                                      39.0
```

	max	std
trip_creation_time	2018-10-03 23:59:42.701692	NaN
od_total_time	7898.55	658.868223
start_scan_to_end_scan	7898.0	658.705957
actual_distance_to_destination	2186.531787	305.388147
actual_time	6265.0	561.396157
osrm_time	2032.0	271.360995
osrm_distance	2840.081	370.395573
segment_actual_time	6230.0	556.247965
segment_osrm_time	2564.0	314.542047
segment_osrm_distance	3523.6324	416.628374
trip_creation_day	30.0	7.893275
trip_creation_month	10.0	0.325757
trip_creation_year	2018.0	0.0
trip_creation_hour	23.0	7.986553
trip_creation_week	40.0	0.967872

```
# Trips created on the hourly basis
hourly_trip_counts = final_grouped_df.groupby(by =
'trip_creation_hour')['trip_uuid'].count()
hourly_trip_counts.sort_values(ascending = False).head()
```

```
trip_creation_hour
22    1125
23    1107
20    1082
0      994
21     873
Name: trip_uuid, dtype: int64
```

```
# Trips created on weekly basis
weekly_trip_counts = final_grouped_df.groupby(by =
```

```
'trip_creation_week')['trip_uuid'].count()
weekly_trip_counts.sort_values(ascending = False).head()
```

```
trip_creation_week
38    5004
39    4417
37    3608
40    1788
Name: trip_uuid, dtype: int64
```

```
# Trips created on day by day basis
daywise_trip_counts = final_grouped_df.groupby(by =
'trip_creation_day')['trip_uuid'].count()
daywise_trip_counts.sort_values(ascending = False).head()
```

```
trip_creation_day
18    791
15    783
13    750
12    747
22    740
Name: trip_uuid, dtype: int64
```

```
# Trips created on different days of the month
monthly_trip_counts = final_grouped_df.groupby(by =
'trip_creation_month')['trip_uuid'].count()
monthly_trip_counts.sort_values(ascending = False).head()
```

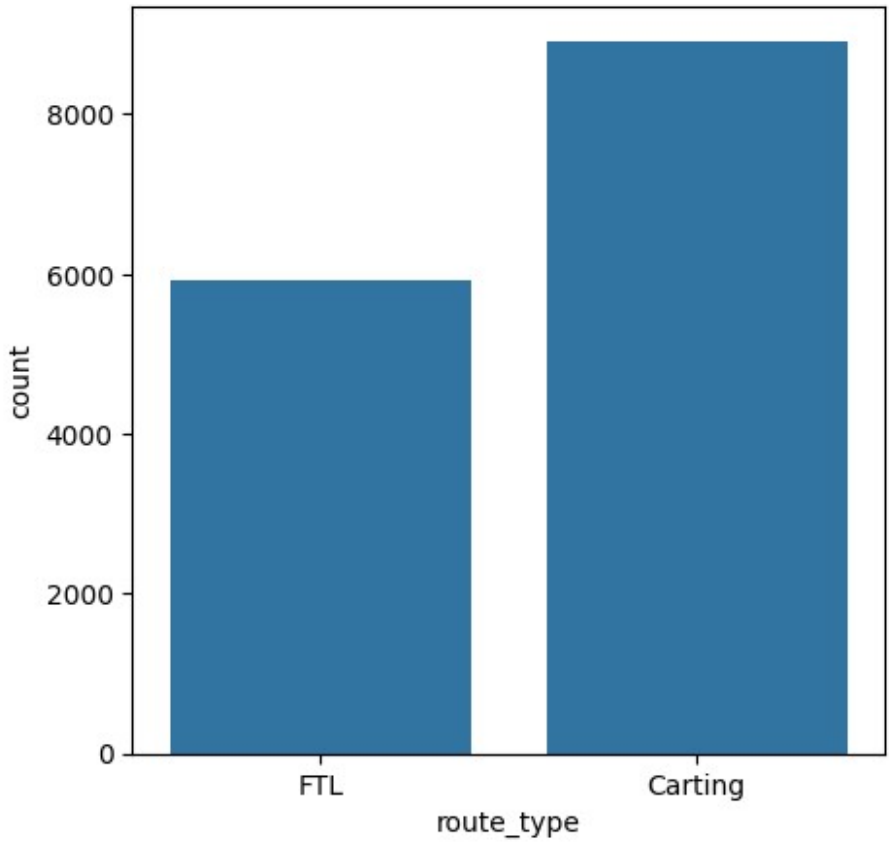
```
trip_creation_month
9     13029
10     1788
Name: trip_uuid, dtype: int64
```

```
# Distribution of trip data
data_count = final_grouped_df.groupby(by = 'data')
['trip_uuid'].count()
data_count
```

```
data
test          4163
training    10654
Name: trip_uuid, dtype: int64
```

```
plt.figure(figsize=(5,5))
plt.pie(data_count,labels=data_count.index, autopct='%1.1f%%')
plt.show()
```


```
plt.figure(figsize=(5,5))
sns.countplot(data = final_grouped_df,x='route_type')
plt.show()
```




```
# Distribution of route type of the given orders
final_grouped_df.groupby(by = 'route_type')['trip_uuid'].count()
```

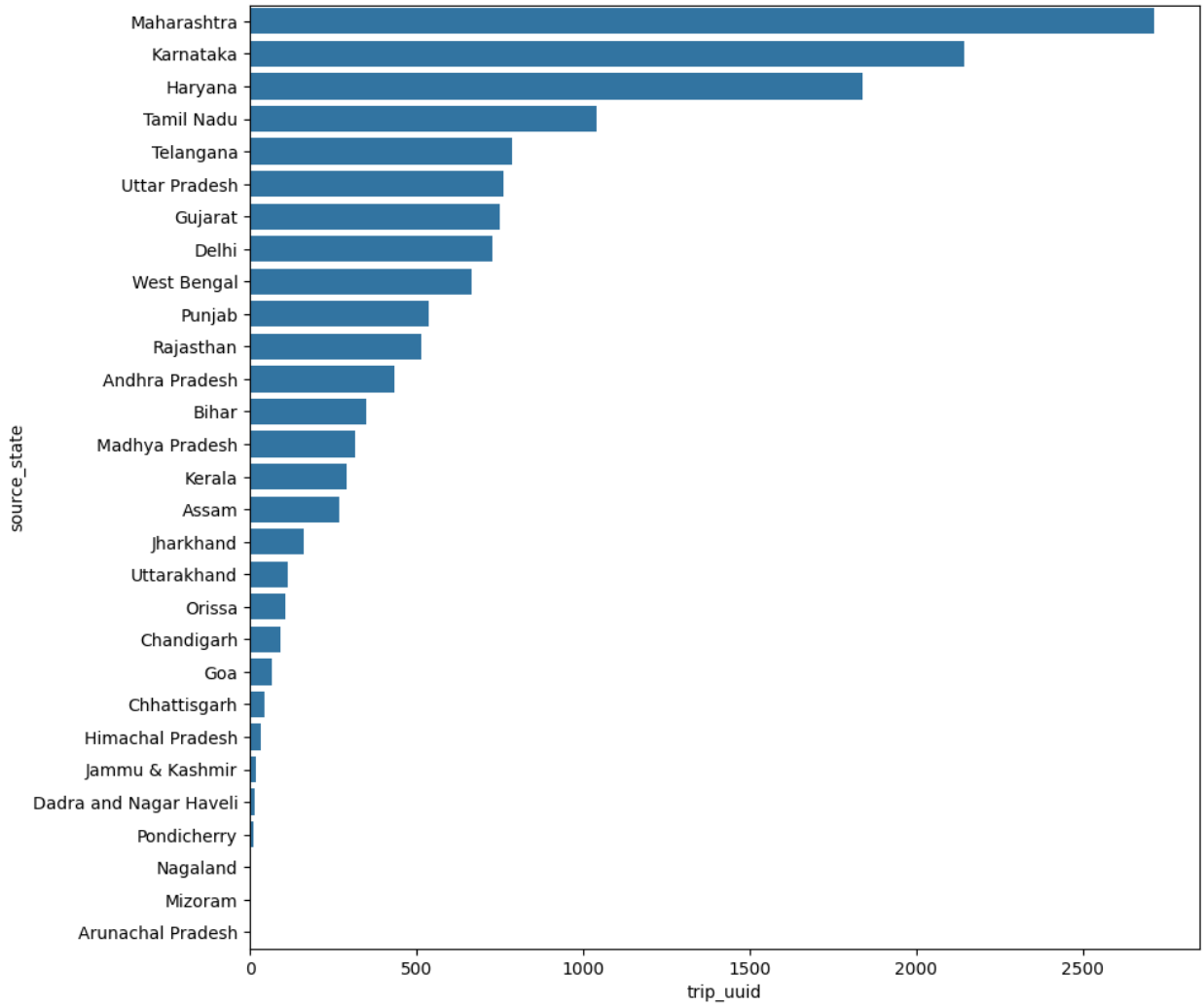
```
route_type
Carting    8908
FTL        5909
Name: trip_uuid, dtype: int64
```

```
# Distribution of number of trips created from different states
source_state_count = final_grouped_df.groupby(by = 'source_state')
['trip_uuid'].count().to_frame().reset_index()
df_state_sort =
source_state_count.sort_values(by='trip_uuid',ascending = False)
df_state_sort.head()
```

```
   source_state  trip_uuid
17  Maharashtra       2714
14    Karnataka       2143
10      Haryana       1838
24   Tamil Nadu       1039
25    Telangana        785
```

```
# Barplot of trips created from different states
plt.figure(figsize = (10, 10))
```

```
sns.barplot(data = df_state_sort,
            x = df_state_sort['trip_uuid'],
            y = df_state_sort['source_state'])
plt.show()
```




```
# Distribution based on the number of trips created to different
cities
destination_city_count = final_grouped_df.groupby(by =
'destination_city')['trip_uuid'].count().to_frame().reset_index()
df_city_sort =
destination_city_count.sort_values(by='trip_uuid',ascending = False)
df_city_sort.head()
```

	destination_city	trip_uuid
515	Mumbai	1548
96	Bengaluru	975
282	Gurgaon	936

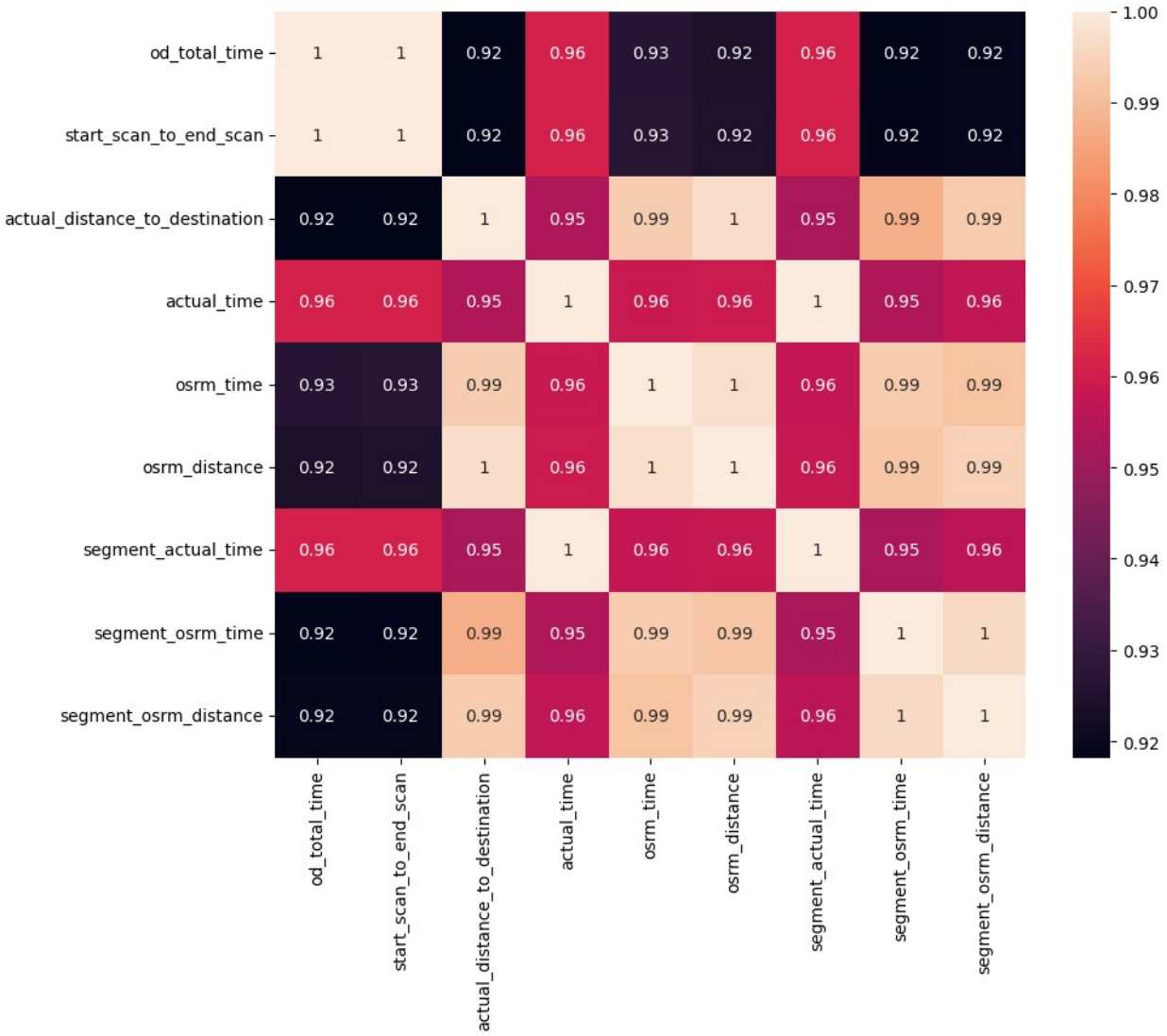
```
200            Delhi        779
163          Chennai        595
```

```
numerical_columns = ['od_total_time', 'start_scan_to_end_scan',
'actual_distance_to_destination',
                     'actual_time', 'osrm_time', 'osrm_distance',
'segment_actual_time',
                     'segment_osrm_time', 'segment_osrm_distance']
df_corr = final_grouped_df[numerical_columns].corr()
df_corr
```

```
                                od_total_time  start_scan_to_end_scan
\
od_total_time                        1.000000                0.999999

start_scan_to_end_scan               0.999999                1.000000

actual_distance_to_destination       0.918222                0.918308

actual_time                          0.961094                0.961147

osrm_time                            0.926516                0.926571

osrm_distance                        0.924219                0.924299

segment_actual_time                  0.961119                0.961171

segment_osrm_time                    0.918490                0.918561

segment_osrm_distance                0.919199                0.919291


                                actual_distance_to_destination
actual_time  \
od_total_time                                         0.918222
0.961094
start_scan_to_end_scan                                0.918308
0.961147
actual_distance_to_destination                        1.000000
0.953757
actual_time                                           0.953757
1.000000
osrm_time                                             0.993561
0.958593
osrm_distance                                         0.997264
0.959214
segment_actual_time                                   0.952821
0.999989
segment_osrm_time                                     0.987538
0.953872
segment_osrm_distance                                 0.993061
```

```
0.956967

                                osrm_time  osrm_distance
segment_actual_time  \
od_total_time                    0.926516       0.924219
0.961119
start_scan_to_end_scan           0.926571       0.924299
0.961171
actual_distance_to_destination   0.993561       0.997264
0.952821
actual_time                      0.958593       0.959214
0.999989
osrm_time                        1.000000       0.997580
0.957765
osrm_distance                    0.997580       1.000000
0.958353
segment_actual_time              0.957765       0.958353
1.000000
segment_osrm_time                0.993259       0.991798
0.953039
segment_osrm_distance            0.991608       0.994710
0.956106

                                segment_osrm_time
segment_osrm_distance
od_total_time                            0.918490
0.919199
start_scan_to_end_scan                   0.918561
0.919291
actual_distance_to_destination           0.987538
0.993061
actual_time                              0.953872
0.956967
osrm_time                                0.993259
0.991608
osrm_distance                            0.991798
0.994710
segment_actual_time                      0.953039
0.956106
segment_osrm_time                        1.000000
0.996092
segment_osrm_distance                    0.996092
1.000000
```

```
plt.figure(figsize = (10,8))
sns.heatmap(data = df_corr, annot = True)
plt.show()
```



Compare the difference between od_total_time and start_scan_to_end_scan.Doing hypothesis testing/ Visual analysis to check.

STEP-1 : Set up Null Hypothesis

Null Hypothesis (H0) - od_total_time (Total Trip Time) and start_scan_to_end_scan (Expected total trip time) are same. Alternate Hypothesis (HA) - od_total_time and start_scan_to_end_scan are different.

STEP-2 : Checking for basic assumpitons for the hypothesis

- Checking histplot to confirm whether its following normal distribution or not
- Distribution check using QQ Plot
- Using shapiro wilks test too check whether its normally distributed or not
- Using Boxcox method to transform the distribution in to normal one and checking again with shapiro to find whether the transformed data follows normal distribution or not
- Homogeneity of Variances using Levene's test

STEP-3: Define Test statistics

If the assumptions of T Test are met then we can proceed performing T Test for independent samples else we will perform the non parametric test equivalent to T Test for independent sample i.e., Mann-Whitney U rank test for two independent samples.

STEP-4: Compute the p-value and fix value of alpha. We set our alpha to be 0.05

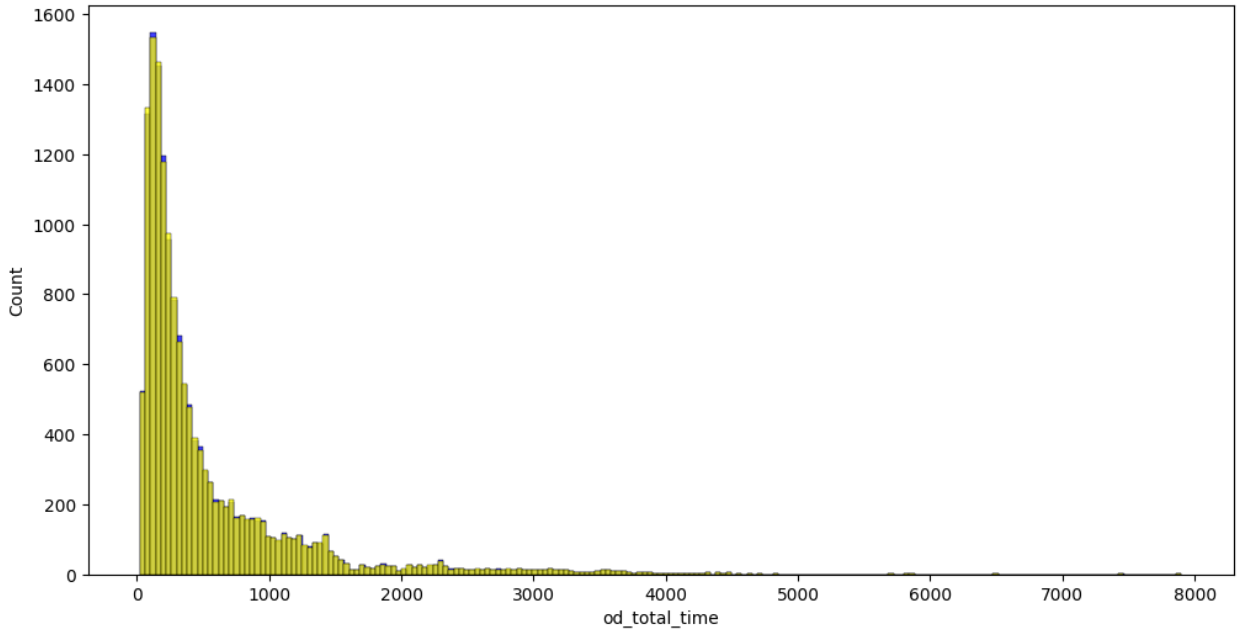
STEP-5: Compare p-value and alpha. Based on p-value, we will accept or reject H0.

p-val > alpha : Accept H0 p-val < alpha : Reject H0

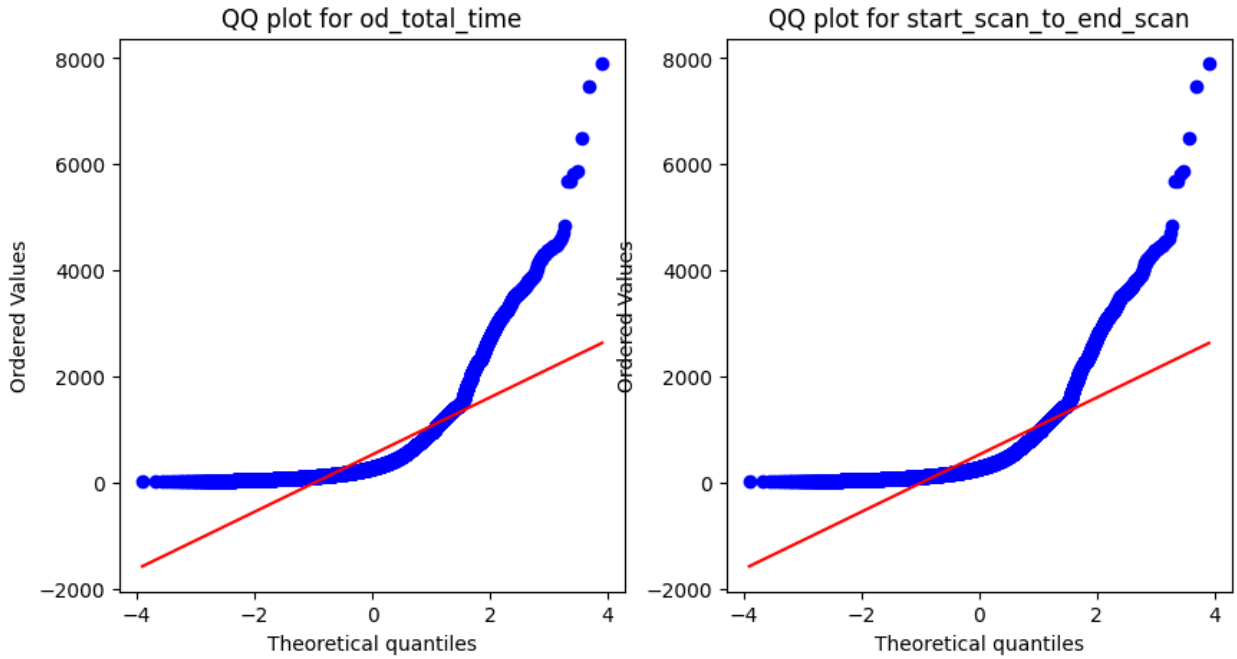
```
# Comparing the difference between od_total_time and
start_scan_to_end_scan. Doing hypothesis testing/ Visual analysis to
check.

final_grouped_df[['od_total_time',
'start_scan_to_end_scan']].describe()
```

```
       od_total_time  start_scan_to_end_scan
count   14817.000000            14817.000000
mean      531.697630              530.810016
std       658.868223              658.705957
min        23.460000               23.000000
25%       149.930000              149.000000
50%       280.770000              280.000000
75%       638.200000              637.000000
max      7898.550000             7898.000000
```

```
plt.figure(figsize = (12, 6))
sns.histplot(final_grouped_df['od_total_time'],color = 'blue')
sns.histplot(final_grouped_df['start_scan_to_end_scan'],color =
'yellow')
plt.show()
```




```
import scipy.stats as stats

plt.figure(figsize = (10, 5))
plt.subplot(1, 2, 1)
stats.probplot(final_grouped_df['od_total_time'], plot = plt, dist =
'norm')
plt.title('QQ plot for od_total_time')
plt.subplot(1, 2, 2)
stats.probplot(final_grouped_df['start_scan_to_end_scan'], plot = plt,
dist = 'norm')
plt.title('QQ plot for start_scan_to_end_scan')
plt.show()
```




```
test_stat, p_value =
stats.shapiro(final_grouped_df['od_total_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 3.371823128416735e-71
The sample does not follow normal distribution
```

```
test_stat, p_value =
stats.shapiro(final_grouped_df['start_scan_to_end_scan'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 6.77853141775857e-72
The sample does not follow normal distribution
```

```
transformed_od_total_time =
stats.boxcox(final_grouped_df['od_total_time'])[0]
test_stat, p_value = stats.shapiro(transformed_od_total_time)
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 7.500440013180809e-25
The sample does not follow normal distribution

/var/folders/qv/81b4b3cn3s750k7zsnnfhvyw0000gp/T/
ipykernel_4679/1962516242.py:2: UserWarning: scipy.stats.shapiro: For
N > 5000, computed p-value may not be accurate. Current N is 14817.
  test_stat, p_value = stats.shapiro(transformed_od_total_time)
```

```
transformed_start_scan_to_end_scan =
stats.boxcox(final_grouped_df['start_scan_to_end_scan'])[0]
test_stat, p_value = stats.shapiro(transformed_start_scan_to_end_scan)
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 1.056337267579965e-24
The sample does not follow normal distribution

/var/folders/qv/81b4b3cn3s750k7zsnnfhvyw0000gp/T/
ipykernel_4679/2971449660.py:2: UserWarning: scipy.stats.shapiro: For
N > 5000, computed p-value may not be accurate. Current N is 14817.
  test_stat, p_value =
stats.shapiro(transformed_start_scan_to_end_scan)
```

```
test_stat, p_value = stats.levene(final_grouped_df['od_total_time'],
final_grouped_df['start_scan_to_end_scan'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples do not have  Homogenous Variance')
else:
    print('The samples have Homogenous Variance ')
```

```
p-value 0.9668007217581142
The samples have Homogenous Variance
```

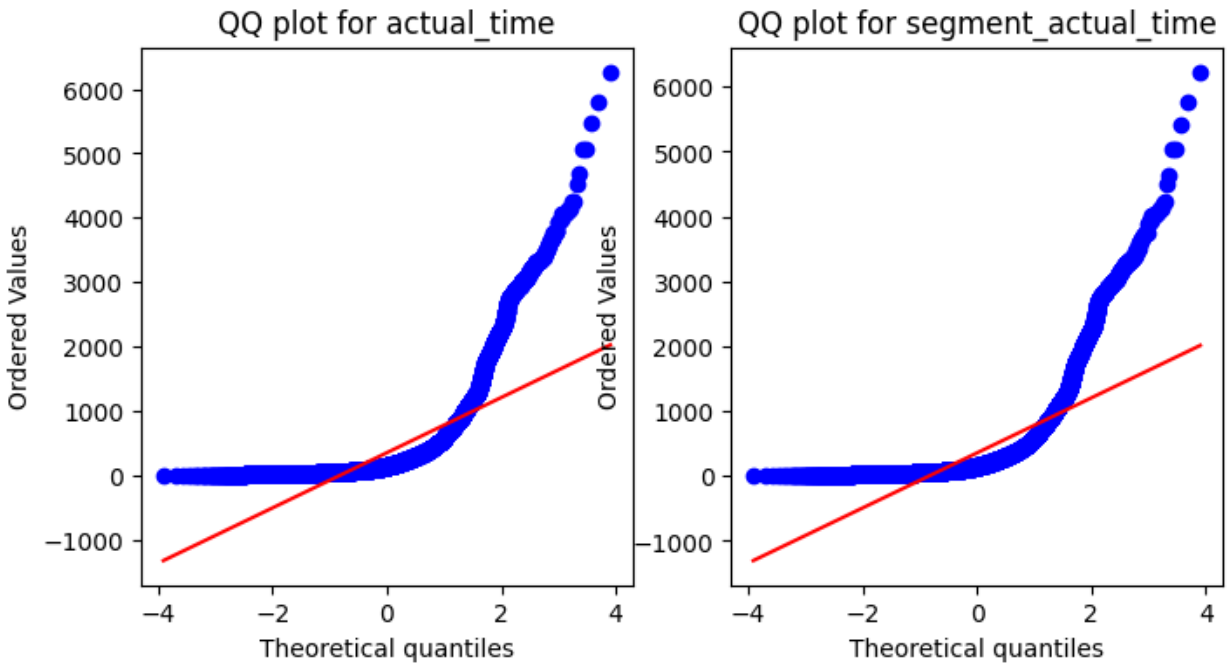
```
test_stat, p_value =
stats.mannwhitneyu(final_grouped_df['od_total_time'],
final_grouped_df['start_scan_to_end_scan'])
print('P-value :',p_value)
if p_value < 0.05:
    print('The od_total_time and start_scan_to_end_scan are not
similar')
else:
    print('The od_total_time and start_scan_to_end_scan are similar')
```

```
P-value : 0.7815123224221716
The od_total_time and start_scan_to_end_scan are similar
```

```
#Doing hypothesis testing between actual_time aggregated value and
segment actual time

final_grouped_df[['actual_time', 'segment_actual_time']].describe()
```

	actual_time	segment_actual_time
count	14817.000000	14817.000000
mean	357.143754	353.892286
std	561.396157	556.247965
min	9.000000	9.000000
25%	67.000000	66.000000
50%	149.000000	147.000000
75%	370.000000	367.000000
max	6265.000000	6230.000000

```
plt.figure(figsize = (8,4))
plt.subplot(1, 2, 1)
stats.probplot(final_grouped_df['actual_time'], plot = plt, dist =
'norm')
plt.title('QQ plot for actual_time')
plt.subplot(1, 2, 2)
stats.probplot(final_grouped_df['segment_actual_time'], plot = plt,
dist = 'norm')
plt.title('QQ plot for segment_actual_time')
plt.show()
```




```
test_stat, p_value = stats.levene(final_grouped_df['actual_time'],
final_grouped_df['segment_actual_time'])
```

```
print('p-value', p_value)
if p_value < 0.05:
    print('The samples do not have Homogenous variance')
else:
    print('The samples have Homogenous variance ')
```

```
p-value 0.6955022668700895
The samples have Homogenous variance
```

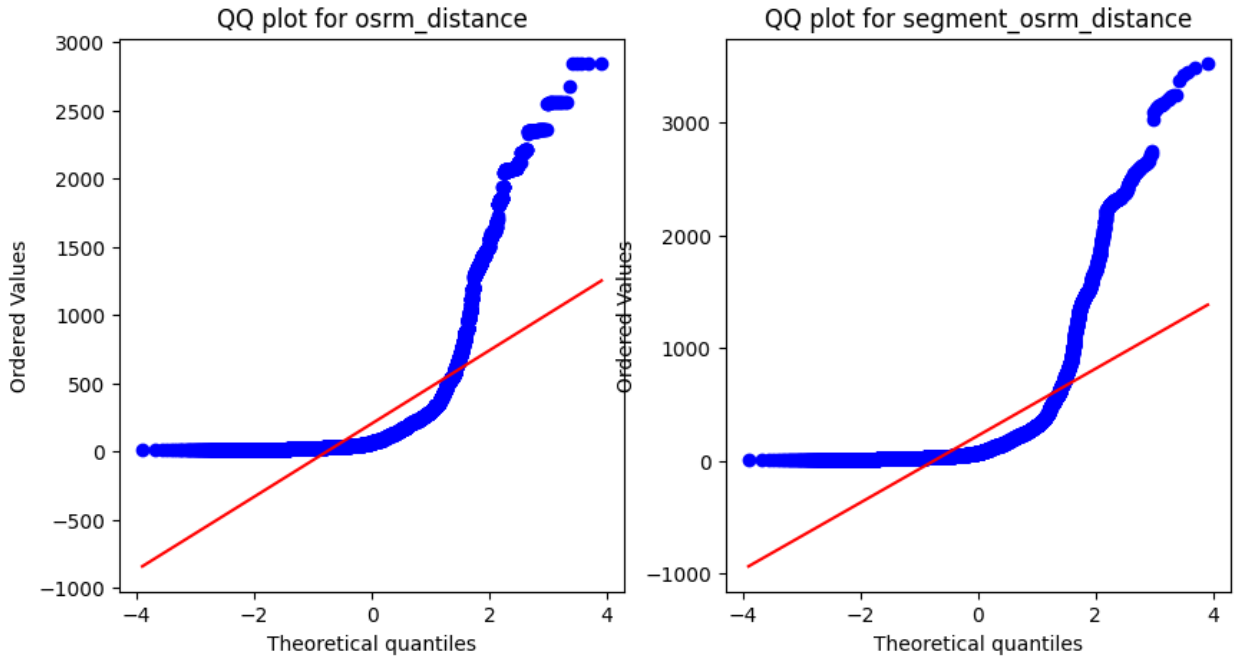
```
test_stat, p_value =
stats.mannwhitneyu(final_grouped_df['actual_time'],
final_grouped_df['segment_actual_time'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples are not similar')
else:
    print('The samples are similar ')
```

```
p-value 0.4164235159622476
The samples are similar
```

```
# Doing hypothesis testing between osrm distance and segment osrm
distance

final_grouped_df[['osrm_distance',
'segment_osrm_distance']].describe()
```

	osrm_distance	segment_osrm_distance
count	14817.000000	14817.000000
mean	204.344689	223.201161
std	370.395573	416.628374
min	9.072900	9.072900
25%	30.819200	32.654500
50%	65.618800	70.154400
75%	208.475000	218.802400
max	2840.081000	3523.632400

```
plt.figure(figsize = (10,5))
plt.subplot(1, 2, 1)
stats.probplot(final_grouped_df['osrm_distance'], plot = plt, dist =
'norm')
plt.title('QQ plot for osrm_distance')
plt.subplot(1, 2, 2)
stats.probplot(final_grouped_df['segment_osrm_distance'], plot = plt,
dist = 'norm')
plt.title('QQ plot for segment_osrm_distance')
plt.show()
```




```
test_stat, p_value =
stats.shapiro(final_grouped_df['osrm_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 1.7796634526878648e-78
The sample does not follow normal distribution
```

```
test_stat, p_value =
stats.shapiro(final_grouped_df['segment_osrm_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 7.86723435485056e-79
The sample does not follow normal distribution
```

```
test_stat, p_value = stats.levene(final_grouped_df['osrm_distance'],
final_grouped_df['segment_osrm_distance'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples do not have homogenous variances')
else:
    print('The samples have homogenous variances')
```

```
p-value 0.00020976354422600578
The samples do not have homogenous variances
```

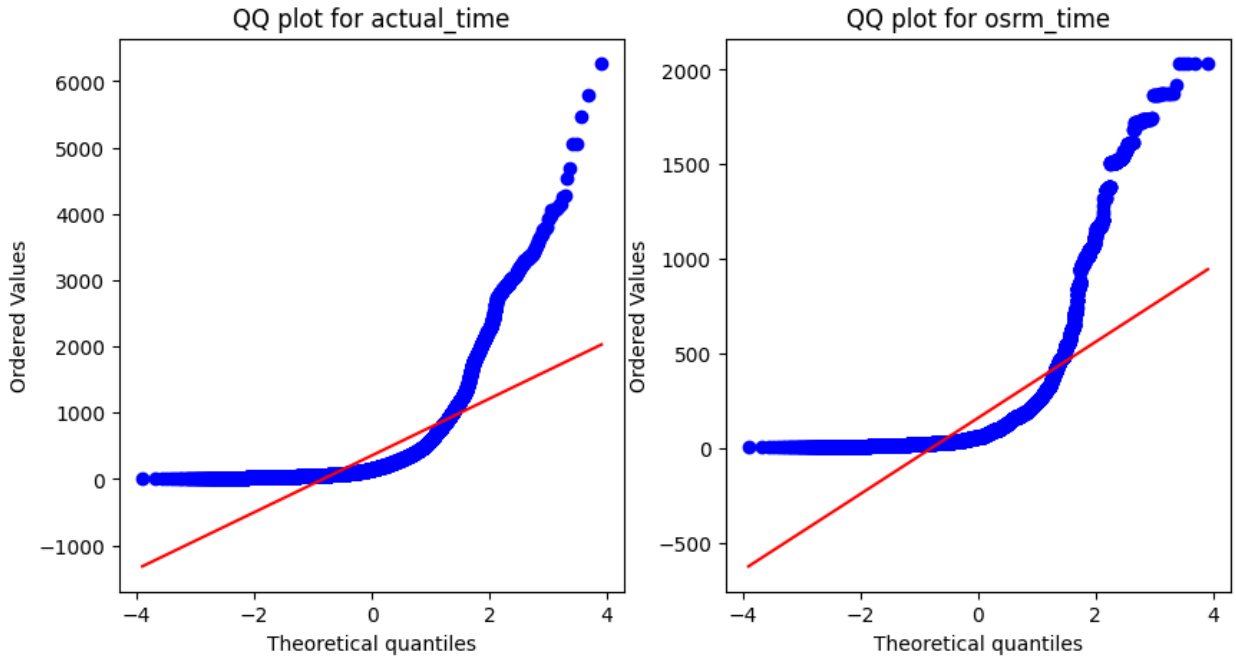
```
test_stat, p_value =
stats.mannwhitneyu(final_grouped_df['osrm_distance'],
final_grouped_df['segment_osrm_distance'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples are not similar')
else:
    print('The samples are similar ')
```

```
p-value 9.511383588276375e-07
The samples are not similar
```

```
# Doing hypothesis tesing between actual time and osrm time

final_grouped_df[['actual_time', 'osrm_time']].describe()
```

	actual_time	osrm_time
count	14817.000000	14817.000000
mean	357.143754	161.384018
std	561.396157	271.360995
min	9.000000	6.000000
25%	67.000000	29.000000
50%	149.000000	60.000000
75%	370.000000	168.000000
max	6265.000000	2032.000000

```
plt.figure(figsize = (10,5))
plt.subplot(1, 2, 1)
stats.probplot(final_grouped_df['actual_time'], plot = plt, dist =
'norm')
plt.title('QQ plot for actual_time')
plt.subplot(1, 2, 2)
stats.probplot(final_grouped_df['osrm_time'], plot = plt, dist =
'norm')
plt.title('QQ plot for osrm_time')
plt.show()
```




```
test_stat, p_value =
stats.shapiro(final_grouped_df['actual_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 7.780361025653668e-77
The sample does not follow normal distribution
```

```
test_stat, p_value =
stats.shapiro(final_grouped_df['osrm_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 3.573637414346663e-78
The sample does not follow normal distribution
```

```
test_stat, p_value = stats.levene(final_grouped_df['actual_time'],
final_grouped_df['osrm_time'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples do not have homogenous variances')
else:
    print('The samples have homogenous variances ')
```

```
p-value 1.871297993683208e-220
The samples do not have homogenous variances
```

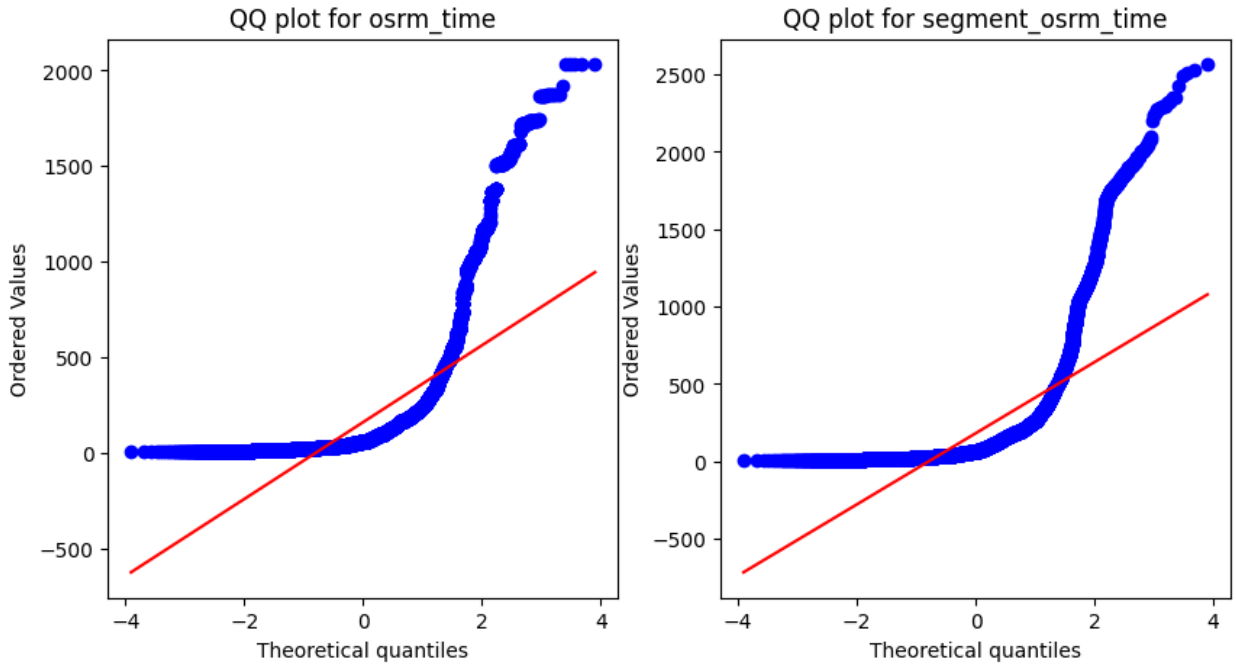
```
test_stat, p_value =
stats.mannwhitneyu(final_grouped_df['actual_time'],final_grouped_df['o
srm_time'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples are not similar')
else:
    print('The samples are similar ')
```

```
p-value 0.0
The samples are not similar
```

```
# Doing hypothesis testing between osrm_time and segment_osrm_time

final_grouped_df[['osrm_time', 'segment_osrm_time']].describe()
```

	osrm_time	segment_osrm_time
count	14817.000000	14817.000000
mean	161.384018	180.949787
std	271.360995	314.542047
min	6.000000	6.000000
25%	29.000000	31.000000
50%	60.000000	65.000000
75%	168.000000	185.000000
max	2032.000000	2564.000000

```
plt.figure(figsize = (10,5))
plt.subplot(1, 2, 1)
stats.probplot(final_grouped_df['osrm_time'], plot = plt, dist =
'norm')
plt.title('QQ plot for osrm_time')
plt.subplot(1, 2, 2)
stats.probplot(final_grouped_df['segment_osrm_time'], plot = plt, dist
= 'norm')
plt.title('QQ plot for segment_osrm_time')
plt.show()
```




```
test_stat, p_value =
stats.shapiro(final_grouped_df['osrm_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

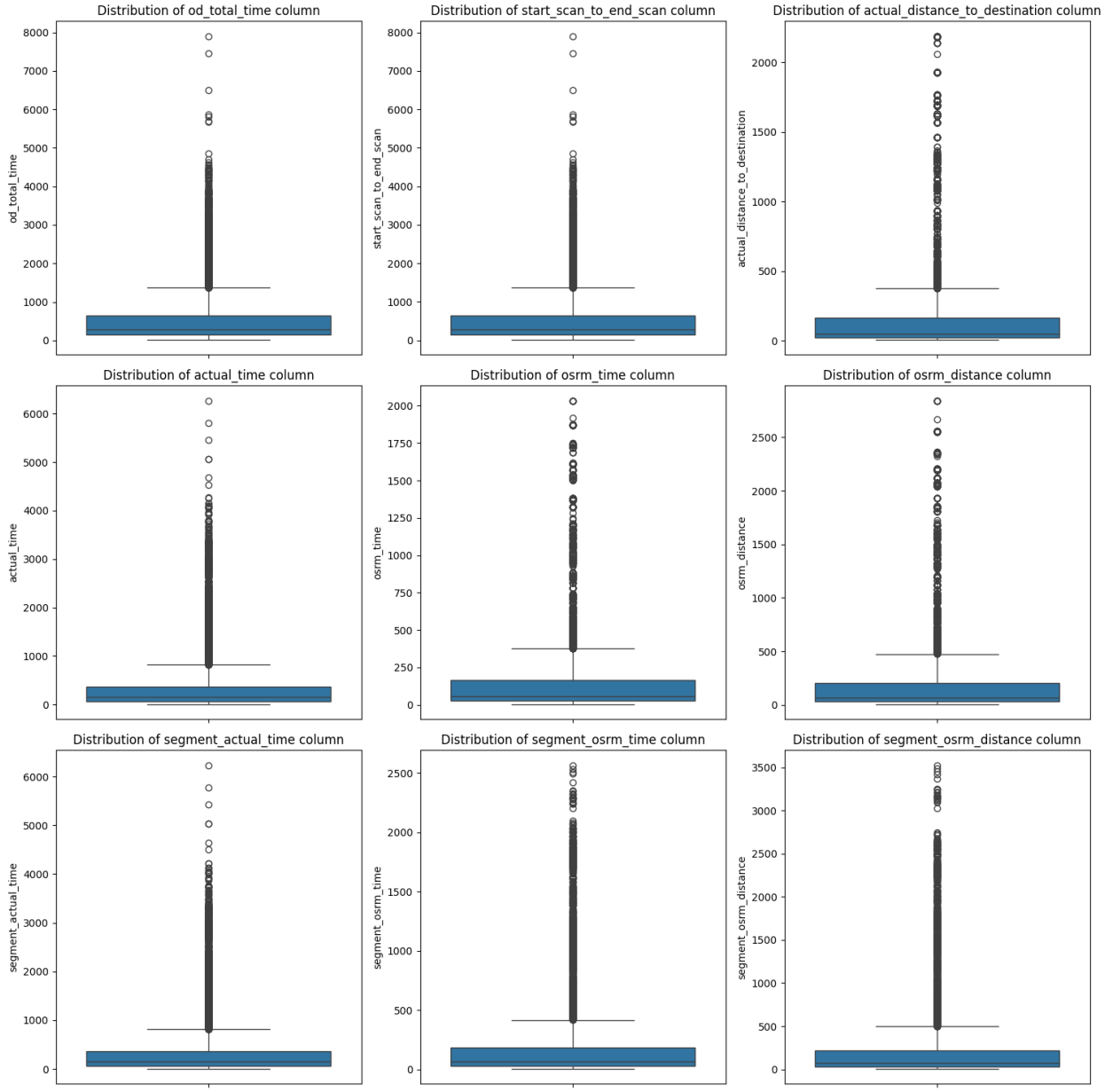
```
p-value 4.6374600126746365e-78
The sample does not follow normal distribution
```

```
test_stat, p_value =
stats.shapiro(final_grouped_df['segment_osrm_time'].sample(5000))
print('p-value', p_value)
if p_value < 0.05:
    print('The sample does not follow normal distribution')
else:
    print('The sample follows normal distribution')
```

```
p-value 3.99615423062762e-79
The sample does not follow normal distribution
```

```
test_stat, p_value = stats.levene(final_grouped_df['osrm_time'],
final_grouped_df['segment_osrm_time'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples do not have homogenous variances')
else:
    print('The samples have homogenous variances')
```

```
p-value 8.3494826690100 88e-08
The samples do not have homogenous variances

test_stat, p_value =
stats.mannwhitneyu(final_grouped_df['osrm_time'],final_grouped_df['seg
ment_osrm_time'])
print('p-value', p_value)
if p_value < 0.05:
    print('The samples are not similar')
else:
    print('The samples are similar ')

p-value 2.2995370859748865e-08
The samples are not similar

numerical_columns = ['od_total_time', 'start_scan_to_end_scan',
'actual_distance_to_destination',
                     'actual_time', 'osrm_time', 'osrm_distance',
'segment_actual_time',
                     'segment_osrm_time', 'segment_osrm_distance']

# Visual analysis of outliers present in all columns using box plot
plt.figure(figsize=(15, 15))
for i in range(len(numerical_columns)):
    plt.subplot(3, 3, i + 1)
    sns.boxplot(final_grouped_df[numerical_columns[i]])
    plt.title(f"Distribution of {numerical_columns[i]} column")

plt.tight_layout()
plt.show()
```




```
# Detecting outliers in numerical columns
for i in numerical_columns:
    Q1 = np.quantile(final_grouped_df[i], 0.25)
    Q3 = np.quantile(final_grouped_df[i], 0.75)
    IQR = Q3 - Q1
    LB = Q1 - 1.5 * IQR
    UB = Q3 + 1.5 * IQR
    outliers = final_grouped_df.loc[(final_grouped_df[i] < LB) |
(final_grouped_df[i] > UB)]
    print('Column :', i)
    print(f'Q1 : {Q1}')
    print(f'Q3 : {Q3}')
```

```
    print(f'IQR : {IQR}')
    print(f'LB : {LB}')
    print(f'UB : {UB}')
    print(f'Number of outliers : {outliers.shape[0]}')
    print('-' * 100)
```

```
Column : od_total_time
Q1 : 149.93
Q3 : 638.2
IQR : 488.27000000000004
LB : -582.4750000000001
UB : 1370.605
Number of outliers : 1266
----------------------------------------------------------------------------------------------------
Column : start_scan_to_end_scan
Q1 : 149.0
Q3 : 637.0
IQR : 488.0
LB : -583.0
UB : 1369.0
Number of outliers : 1267
----------------------------------------------------------------------------------------------------
Column : actual_distance_to_destination
Q1 : 22.83723905859321
Q3 : 164.58320763841138
IQR : 141.74596857981817
LB : -189.78171381113404
UB : 377.2021605081386
Number of outliers : 1449
----------------------------------------------------------------------------------------------------
Column : actual_time
Q1 : 67.0
Q3 : 370.0
IQR : 303.0
LB : -387.5
UB : 824.5
Number of outliers : 1643
----------------------------------------------------------------------------------------------------
Column : osrm_time
Q1 : 29.0
Q3 : 168.0
IQR : 139.0
LB : -179.5
UB : 376.5
Number of outliers : 1517
----------------------------------------------------------------------------------------------------
```

```
------------------------------
Column : osrm_distance
Q1 : 30.8192
Q3 : 208.475
IQR : 177.6558
LB : -235.6645
UB : 474.9587
Number of outliers : 1524
-------------------------------------------------------------------------------
------------------------------
Column : segment_actual_time
Q1 : 66.0
Q3 : 367.0
IQR : 301.0
LB : -385.5
UB : 818.5
Number of outliers : 1643
-------------------------------------------------------------------------------
------------------------------
Column : segment_osrm_time
Q1 : 31.0
Q3 : 185.0
IQR : 154.0
LB : -200.0
UB : 416.0
Number of outliers : 1492
-------------------------------------------------------------------------------
------------------------------
Column : segment_osrm_distance
Q1 : 32.6545
Q3 : 218.8024
IQR : 186.1479
LB : -246.56735000000003
UB : 498.02425000000005
Number of outliers : 1548
-------------------------------------------------------------------------------
------------------------------
```

```
# One-hot encoding of  multiple categorical columns
encoded_df = pd.get_dummies(final_grouped_df, columns=['route_type',
'data'])

encoded_df.head()
```

```
                  trip_uuid source_center destination_center  \
0  trip-153671041653548748  IND209304AAA       IND209304AAA
1  trip-153671042288605164  IND561203AAB       IND561203AAB
2  trip-153671043369099517  IND000000ACB       IND000000ACB
3  trip-153671046011330457  IND400072AAB       IND401104AAA
4  trip-153671052974046625  IND583101AAA       IND583119AAA
```

```
          trip_creation_time                          source_name  \
0 2018-09-12 00:00:16.535741  Kanpur_Central_H_6 (Uttar Pradesh)
1 2018-09-12 00:00:22.886430   Doddablpur_ChikaDPP_D (Karnataka)
2 2018-09-12 00:00:33.691250       Gurgaon_Bilaspur_HB (Haryana)
3 2018-09-12 00:01:00.113710            Mumbai Hub (Maharashtra)
4 2018-09-12 00:02:09.740725              Bellary_Dc (Karnataka)

                     destination_name  od_total_time
start_scan_to_end_scan  \
0  Kanpur_Central_H_6 (Uttar Pradesh)        2260.11
2259.0
1   Doddablpur_ChikaDPP_D (Karnataka)         181.61
180.0
2       Gurgaon_Bilaspur_HB (Haryana)        3934.36
3933.0
3      Mumbai_MiraRd_IP (Maharashtra)         100.49
100.0
4       Sandur_WrdN1DPP_D (Karnataka)         718.34
717.0

   actual_distance_to_destination  actual_time  ...  destination_place
\
0                      824.732854       1562.0  ...        Central_H_6

1                       73.186911        143.0  ...         ChikaDPP_D

2                     1927.404273       3347.0  ...        Bilaspur_HB

3                       17.175274         59.0  ...          MiraRd_IP

4                      127.448500        341.0  ...         WrdN1DPP_D


   trip_creation_date  trip_creation_day  trip_creation_month  \
0          2018-09-12                 12                    9
1          2018-09-12                 12                    9
2          2018-09-12                 12                    9
3          2018-09-12                 12                    9
4          2018-09-12                 12                    9

   trip_creation_year destination_city route_type_Carting
route_type_FTL  \
0                2018           Kanpur               False
True
1                2018       Doddablpur                True
False
2                2018          Gurgaon               False
True
3                2018           Mumbai                True
```

```
False
4                2018           Sandur               False
True

  data_test data_training
0     False          True
1     False          True
2     False          True
3     False          True
4     False          True

[5 rows x 29 columns]
```

```
# Normalize/ Standardize the numerical features using MinMaxScaler or
StandardScaler.

from sklearn.preprocessing import StandardScaler

# List of numerical columns to scale
numerical_columns = ['od_total_time', 'start_scan_to_end_scan',
'actual_distance_to_destination','actual_time', 'osrm_time',
'osrm_distance', 'segment_actual_time',
                     'segment_osrm_time', 'segment_osrm_distance']

scaler = StandardScaler()

final_grouped_df[numerical_columns] =
scaler.fit_transform(final_grouped_df[numerical_columns])
final_grouped_df.head()
```

```
                  trip_uuid source_center destination_center      data
\
0  trip-153671041653548748  IND209304AAA       IND209304AAA  training

1  trip-153671042288605164  IND561203AAB       IND561203AAB  training

2  trip-153671043369099517  IND000000ACB       IND000000ACB  training

3  trip-153671046011330457  IND400072AAB       IND401104AAA  training

4  trip-153671052974046625  IND583101AAA       IND583119AAA  training


  route_type          trip_creation_time
source_name  \
0        FTL  2018-09-12 00:00:16.535741  Kanpur_Central_H_6 (Uttar
Pradesh)
1    Carting  2018-09-12 00:00:22.886430   Doddablpur_ChikaDPP_D
(Karnataka)
2        FTL  2018-09-12 00:00:33.691250       Gurgaon_Bilaspur_HB
(Haryana)
```

```
3    Carting  2018-09-12 00:01:00.113710            Mumbai Hub
(Maharashtra)
4        FTL  2018-09-12 00:02:09.740725              Bellary_Dc
(Karnataka)

                     destination_name  od_total_time
start_scan_to_end_scan  \
0  Kanpur_Central_H_6 (Uttar Pradesh)       2.623394
2.623702
1   Doddablpur_ChikaDPP_D (Karnataka)      -0.531365               -
0.532593
2       Gurgaon_Bilaspur_HB (Haryana)       5.164579
5.165134
3      Mumbai_MiraRd_IP (Maharashtra)      -0.654489               -
0.654047
4       Sandur_WrdN1DPP_D (Karnataka)       0.283287
0.282670

   actual_distance_to_destination  actual_time  osrm_time
osrm_distance  \
0                        2.162092     2.146251   2.047585
2.124848
1                       -0.298944    -0.381461  -0.344144      -
0.321920
2                        5.772935     5.325931   5.817598
5.804050
3                       -0.482362    -0.531093  -0.539462      -
0.498578
4                       -0.121257    -0.028757  -0.163566      -
0.155387

   segment_actual_time  segment_osrm_time  segment_osrm_distance
0             2.146791           2.629468               2.633784
1            -0.382742          -0.368643              -0.333670
2             5.310954           5.595785               5.573660
3            -0.530163          -0.524430              -0.488040
4            -0.024976          -0.209676              -0.183405
```

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
final_grouped_df[numerical_columns] =
scaler.fit_transform(final_grouped_df[numerical_columns])

final_grouped_df.head()
```

```
                 trip_uuid source_center destination_center      data
\
0  trip-153671041653548748  IND209304AAA       IND209304AAA  training
```

```
1  trip-153671042288605164  IND561203AAB       IND561203AAB  training

2  trip-153671043369099517  IND000000ACB       IND000000ACB  training

3  trip-153671046011330457  IND400072AAB       IND401104AAA  training

4  trip-153671052974046625  IND583101AAA       IND583119AAA  training


  route_type          trip_creation_time
source_name  \
0        FTL  2018-09-12 00:00:16.535741  Kanpur_Central_H_6 (Uttar
Pradesh)
1    Carting  2018-09-12 00:00:22.886430   Doddablpur_ChikaDPP_D
(Karnataka)
2        FTL  2018-09-12 00:00:33.691250       Gurgaon_Bilaspur_HB
(Haryana)
3    Carting  2018-09-12 00:01:00.113710            Mumbai Hub
(Maharashtra)
4        FTL  2018-09-12 00:02:09.740725              Bellary_Dc
(Karnataka)

                     destination_name  od_total_time
start_scan_to_end_scan  \
0  Kanpur_Central_H_6 (Uttar Pradesh)       0.284016
0.283937
1   Doddablpur_ChikaDPP_D (Karnataka)       0.020082
0.019937
2       Gurgaon_Bilaspur_HB (Haryana)       0.496617
0.496508
3     Mumbai_MiraRd_IP (Maharashtra)       0.009781
0.009778
4       Sandur_WrdN1DPP_D (Karnataka)       0.088238
0.088127

   actual_distance_to_destination  actual_time  osrm_time
osrm_distance  \
0                        0.374613     0.248242   0.350938
0.346972
1                        0.029476     0.021419   0.030602
0.026859
2                        0.880999     0.533568   0.855874
0.828325
3                        0.003753     0.007992   0.004442
0.003747
4                        0.054395     0.053069   0.054788
0.048647

   segment_actual_time  segment_osrm_time  segment_osrm_distance
0             0.247388           0.391712               0.373134
```

```
1               0.021218            0.023065                0.021373
2               0.530301            0.756450                0.721625
3               0.008037            0.003909                0.003074
4               0.053207            0.042611                0.039185
```

Inferences:

- There are about 14817 unique trip IDs, 1508 unique source centers, 1481 unique destination_centers.
- Testing type data is more than the training type data.
- Carting route type is popular than FTL
- The number of trips start increasing after the noon, becomes maximum at 10 P.M
- Highest number of trips are created in the 38th week.
- Number of orders sourced from the states like Maharashtra, Karnataka, Haryana, Tamil Nadu, Telangana are high.
- Maximum number of trips destination Maharashtra state followed by Karnataka, Haryana. That means that the number of orders placed in these states is significantly high.
- Maximum number of trips ended in Mumbai city followed by Bengaluru, Gurgaon, Delhi and so on
- Features like start_scan_to_end_scan and od_total_time(aggregated data) are statistically similar.
- Features actual_time & osrm_time are statitically different.
- Features start_scan_to_end_scan and segment_actual_time are statistically similar.
- Features osrm_distance and segment_osrm_distance are statistically different from each other.
- Both the osrm_time & segment_osrm_time are statistically different.

Recommendations:-

- Most of the orders are coming from/reaching to states like Maharashtra, Karnataka, Haryana.These routes can be specially focused up on to maintain an edge over competitors.
- Other states like North East territories need to be focused up on as they are not popular routes.These areas should be given special attention by analysing types of customers and orders.
- osrm_time and actual_time are statistically different.This needs to be checked up on as this is an important parameter to gain customer satisfaction.
- The osrm distance and actual distance covered are also not same,checking whether the error is because of techincal glitch need to be done.
- customer satisfaction and greivance redressal are to be given priority,proper feed back mechanism and analysing customer reviews are to be done.