Delhivery - Feature Engineering

May 29, 2024

[26]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from scipy.stats import stats
```

[6]:
```
df = pd.read_csv("/Users/senth/Downloads/delhivery_data.csv")
```

[7]:
```
df.head()
```

[7]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_center	source_name	destination_center	destination_name
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	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]
```

1 Understanding Shape and Structure of Data

[8]:
```
df.shape
```

[8]:
```
(144867, 24)
```

[10]:
```
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
```

[11]:
```
df.isna().sum()
```

[11]:
```
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
```

[13]:
```
df.dtypes
```

[13]:
```
data                              object
trip_creation_time                object
route_schedule_uuid               object
route_type                        object
trip_uuid                         object
source_center                     object
source_name                       object
destination_center                object
destination_name                  object
od_start_time                     object
od_end_time                       object
start_scan_to_end_scan           float64
is_cutoff                           bool
cutoff_factor                      int64
cutoff_timestamp                  object
actual_distance_to_destination   float64
actual_time                      float64
osrm_time                        float64
osrm_distance                    float64
factor                           float64
segment_actual_time              float64
segment_osrm_time                float64
segment_osrm_distance            float64
segment_factor                   float64
dtype: object
```

[14]:
```
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"])
```

[16]:
```
df["trip_creation_time"].dt.month_name().value_counts()
```

[16]:
```
September    127349
October       17518
Name: trip_creation_time, dtype: int64
```

[18]:
```
df["trip_creation_time"].dt.year.value_counts()
```

[18]:
```
2018    144867
Name: trip_creation_time, dtype: int64
```

[19]:
```
df["trip_creation_time"].dt.day_name().value_counts()
```

[19]:
```
Wednesday    26732
Thursday     20481
Friday       20242
Tuesday      19961
Saturday     19936
Monday       19645
Sunday       17870
Name: trip_creation_time, dtype: int64
```

[20]:
```
df.nunique()
```

[20]:
```
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
```

[21]:
```
df.describe()
```

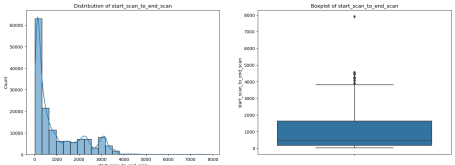
[21]:
```
       start_scan_to_end_scan  cutoff_factor  actual_distance_to_destination  \
count           144867.000000  144867.000000                   144867.000000
mean               961.262986     232.926567                      234.073372
std               1037.012769     344.755577                      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
```

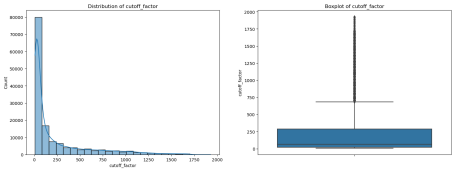
1. There are total 14817 different trips of data available
2. There are 1508 unique source_center
3. There are 1481 unique destination_center
4. There are total 1504 delivery routes

2 Visual Analysis

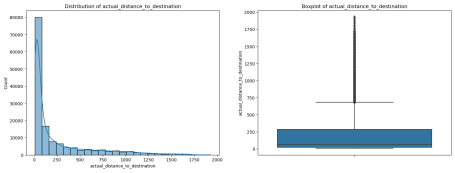
[23]:
```
num_vars = df.select_dtypes(include=np.number).columns.tolist()

fig, ax = plt.subplots(nrows=11, ncols=2, figsize=(18, 80))

for i in range(len(num_vars)):

    sns.histplot(x=df[num_vars[i]], kde=True, bins = 25, ax=ax[i, 0])
    ax[i, 0].set_title(f"Distribution of {num_vars[i]}")

    sns.boxplot(y = df[num_vars[i]], ax=ax[i, 1], data=df)
    ax[i, 1].set_title(f"Boxplot of {num_vars[i]}")

plt.show()
```


Distribution of start_scan_to_end_scan
Count
start_scan_to_end_scan

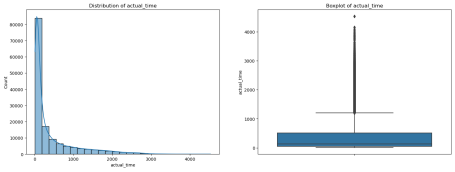
Boxplot of start_scan_to_end_scan
start_scan_to_end_scan


Distribution of cutoff_factor
Count
cutoff_factor

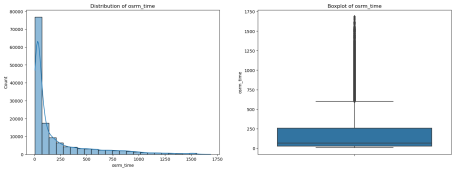
Boxplot of cutoff_factor
cutoff_factor


Distribution of actual_distance_to_destination
Count
actual_distance_to_destination

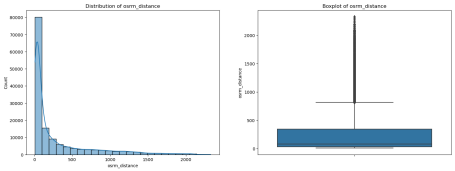
Boxplot of actual_distance_to_destination
actual_distance_to_destination


Distribution of actual_time
Count
actual_time

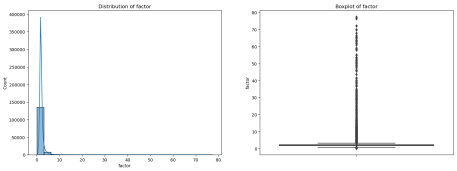
Boxplot of actual_time
actual_time


Distribution of osrm_time
Count
osrm_time

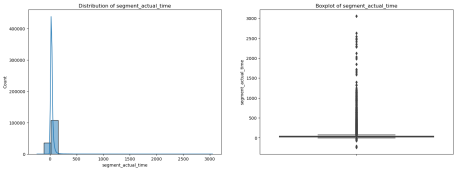
Boxplot of osrm_time
osrm_time


Distribution of osrm_distance
Count
osrm_distance

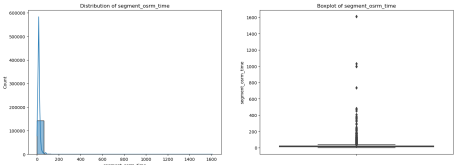
Boxplot of osrm_distance
osrm_distance


Distribution of factor
Count
factor

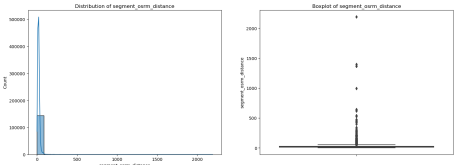
Boxplot of factor
factor


Distribution of segment_actual_time
Count
segment_actual_time

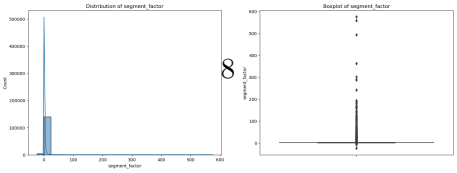
Boxplot of segment_actual_time
segment_actual_time


Distribution of segment_osrm_time
Count
segment_osrm_time

Boxplot of segment_osrm_time
segment_osrm_time


Distribution of segment_osrm_distance
Count
segment_osrm_distance

Boxplot of segment_osrm_distance
segment_osrm_distance


Distribution of segment_factor
Count
segment_factor

Boxplot of segment_factor
segment_factor

3 Feature Creation

[27]:
```
df["source_city"] = df["source_name"].str.split(" ",n=1,expand=True)[0].str.
 ↪split("_",n=1,expand=True)[0]
df["source_state"] = df["source_name"].str.split(" ",n=1,expand=True)[1].str.
 ↪replace("(","").str.replace(")","")

df["destination_city"] = df["destination_name"].str.split("␣
 ↪",n=1,expand=True)[0].str.split("_",n=1,expand=True)[0]
df["destination_state"] = df["destination_name"].str.split("␣
 ↪",n=1,expand=True)[1].str.replace("(","").str.replace(")","")

df["source_pincode"] = df["source_center"].apply(lambda x : x[3:9] )
df["destination_pincode"] = df["destination_center"].apply(lambda x : x[3:9] )
```

[28]:
```
df["time_taken_btwn_odstart_and_od_end"] =␣
 ↪((df["od_end_time"]-df["od_start_time"])/pd.Timedelta(1,unit="hour"))
```

[29]:
```
df["start_scan_to_end_scan"] = df["start_scan_to_end_scan"]/60
df["actual_time"] = df["actual_time"]/60
df["osrm_time"] = df["osrm_time"]/60
df["segment_actual_time"] = df["segment_actual_time"]/60
df["segment_osrm_time"] = df["segment_osrm_time"]/60
```

[30]:
```
df.head()
```

[30]:

	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  … segment_osrm_time  segment_osrm_distance  \
0 2018-09-20 03:21:32.418600  …          0.183333                11.9653
1 2018-09-20 03:21:32.418600  …          0.150000                 9.7590
2 2018-09-20 03:21:32.418600  …          0.116667                10.8152
3 2018-09-20 03:21:32.418600  …          0.200000                13.0224
4 2018-09-20 03:21:32.418600  …          0.083333                 3.9153

   segment_factor  source_city source_state  destination_city  \
0        1.272727        Anand      Gujarat          Khambhat
1        1.111111        Anand      Gujarat          Khambhat
2        2.285714        Anand      Gujarat          Khambhat
3        1.750000        Anand      Gujarat          Khambhat
4        1.200000        Anand      Gujarat          Khambhat

   destination_state  source_pincode  destination_pincode  \
0           Gujarat          388121               388620
1           Gujarat          388121               388620
2           Gujarat          388121               388620
3           Gujarat          388121               388620
4           Gujarat          388121               388620

   time_taken_btwn_odstart_and_od_end
0                            1.436894
1                            1.436894
2                            1.436894
3                            1.436894
4                            1.436894

[5 rows x 31 columns]
```

[31]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144867 entries, 0 to 144866
Data columns (total 31 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   data                                144867 non-null  object
```

```
 1   trip_creation_time                  144867 non-null  datetime64[ns]
 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  datetime64[ns]
 10  od_end_time                         144867 non-null  datetime64[ns]
 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
 24  source_city                         144574 non-null  object
 25  source_state                        144574 non-null  object
 26  destination_city                    144606 non-null  object
 27  destination_state                   144606 non-null  object
 28  source_pincode                      144867 non-null  object
 29  destination_pincode                 144867 non-null  object
 30  time_taken_btwn_odstart_and_od_end  144867 non-null  float64
dtypes: bool(1), datetime64[ns](3), float64(11), int64(1), object(15)
memory usage: 33.3+ MB
```

4 Data Cleaning

[34]:
```
df["source_state"] = df["source_state"].replace({"Goa Goa":"Goa",
                            "Layout PC Karnataka":"Karnataka",
                            "Vadgaon Sheri DPC Maharashtra":"Maharashtra",
                            "Pashan DPC Maharashtra":"Maharashtra",
                            "City Madhya Pradesh":"Madhya Pradesh",
                            "02_DPC Uttar Pradesh":"Uttar Pradesh",
                            "Nagar_DC Rajasthan":"Rajasthan",
                            "Alipore_DPC West Bengal":"West Bengal",
                             "Mandakni Madhya Pradesh":"Madhya Pradesh",
                             "West _Dc Maharashtra":"Maharashtra",
                             "DC Rajasthan":"Rajasthan",
                             "MP Nagar Madhya Pradesh":"Madhya Pradesh",
```

```
                                "Antop Hill Maharashtra":"Maharashtra",
                                "Avenue_DPC West Bengal":"West Bengal",
                                "Nagar Uttar Pradesh":"Uttar Pradesh",
                                "Balaji Nagar Maharashtra":"Maharashtra",
                                "Kothanur_L Karnataka":"Karnataka",
                                "Rahatani DPC Maharashtra":"Maharashtra",
                                "Mahim Maharashtra":"Maharashtra",
                                "DC Maharashtra":"Maharashtra",
                                "_NAD Andhra Pradesh":"Andhra Pradesh",
                                                        })
```

[35]:
```
df["destination_state"] = df["destination_state"].replace({"Goa Goa":"Goa",
                               "Layout PC Karnataka":"Karnataka",
                               "Vadgaon Sheri DPC Maharashtra":"Maharashtra",
                               "Pashan DPC Maharashtra":"Maharashtra",
                               "City Madhya Pradesh":"Madhya Pradesh",
                               "02_DPC Uttar Pradesh":"Uttar Pradesh",
                               "Nagar_DC Rajasthan":"Rajasthan",
                               "Alipore_DPC West Bengal":"West Bengal",
                                "Mandakni Madhya Pradesh":"Madhya Pradesh",
                                "West _Dc Maharashtra":"Maharashtra",
                                "DC Rajasthan":"Rajasthan",
                                "MP Nagar Madhya Pradesh":"Madhya Pradesh",
                                "Antop Hill Maharashtra":"Maharashtra",
                                "Avenue_DPC West Bengal":"West Bengal",
                                "Nagar Uttar Pradesh":"Uttar Pradesh",
                                "Balaji Nagar Maharashtra":"Maharashtra",
                                "Kothanur_L Karnataka":"Karnataka",
                                "Rahatani DPC Maharashtra":"Maharashtra",
                                "Mahim Maharashtra":"Maharashtra",
                                "DC Maharashtra":"Maharashtra",
                                "_NAD Andhra Pradesh":"Andhra Pradesh",
                               "Delhi Delhi":"Delhi",
                               "West_Dc Maharashtra":"Maharashtra",
                               "Hub Maharashtra":"Maharashtra"
                                                        })
```

[37]:
```
df["destination_city"].replace({
    "del":"Delhi"
},inplace=True)
df["source_city"].replace({
    "del":"Delhi"
},inplace=True)

df["source_city"].replace({
    "Bangalore":"Bengaluru"
        },inplace=True)
```

```
df["destination_city"].replace({
    "Bangalore":"Bengaluru"
        },inplace=True)
df["destination_city"].replace({
    "AMD":"Ahmedabad"
        },inplace=True)
df["destination_city"].replace({
    "Amdavad":"Ahmedabad"
        },inplace=True)
df["source_city"].replace({
    "AMD":"Ahmedabad"
        },inplace=True)
df["source_city"].replace({
    "Amdavad":"Ahmedabad"
        },inplace=True)
```

[38]:
```
df["source_city_state"] = df["source_city"] + " " + df["source_state"]
df["destination_city_state"] = df["destination_city"] + " " +␣
 ↪df["destination_state"]
```

[39]:
```
df["source_city_state"].nunique()
```

[39]: 1249

[41]:
```
df["destination_city_state"].nunique()
```

[41]: 1242

[42]:
```
df["source_state"].nunique()
```

[42]: 33

[43]:
```
df["destination_state"].nunique()
```

[43]: 32

4.0.1 Dropping Unnecessary columns

[44]:
```
data = df.copy()
```

[45]:
```
data.shape
```

[45]: (144867, 33)

[46]:
```
data.drop(
   ␣
 ↪['source_center',"source_name","destination_center","destination_name","cutoff_timestamp",␣
 ↪"od_end_time","od_start_time"],
```

```
    axis = 1,
    inplace=True
)
```

[47]:
```
data.shape
```

[47]:
```
(144867, 26)
```

5 Merging of rows and aggregation of fields

[48]:
```
actual_time = data.groupby(["trip_uuid",
              "start_scan_to_end_scan"])["actual_time"].max().reset_index().
 ↪groupby("trip_uuid")["actual_time"].sum().reset_index()

actual_time
```

[48]:

	trip_uuid	actual_time
0	trip-153671041653548748	26.033333
1	trip-153671042288605164	2.383333
2	trip-153671043369099517	55.783333
3	trip-153671046011330457	0.983333
4	trip-153671052974046625	5.683333
…	…	…
14812	trip-153861095625827784	1.383333
14813	trip-153861104386292051	0.350000
14814	trip-153861106442901555	4.700000
14815	trip-153861115439069069	4.400000
14816	trip-153861118270144424	4.583333

[14817 rows x 2 columns]

[49]:
```
segment_osrm_time = data[["trip_uuid","segment_osrm_time"]].
 ↪groupby("trip_uuid")["segment_osrm_time"].sum().reset_index()
segment_osrm_time
```

[49]:

	trip_uuid	segment_osrm_time
0	trip-153671041653548748	16.800000
1	trip-153671042288605164	1.083333
2	trip-153671043369099517	32.350000
3	trip-153671046011330457	0.266667
4	trip-153671052974046625	1.916667
…	…	…
14812	trip-153861095625827784	1.033333
14813	trip-153861104386292051	0.183333
14814	trip-153861106442901555	1.466667
14815	trip-153861115439069069	3.683333

```
14816  trip-153861118270144424           1.116667

[14817 rows x 2 columns]
```

[50]:
```
segment_actual_time = data.groupby("trip_uuid")["segment_actual_time"].sum().
 ↪reset_index()
segment_actual_time
```

[50]:
```
                     trip_uuid  segment_actual_time
0      trip-153671041653548748            25.800000
1      trip-153671042288605164             2.350000
2      trip-153671043369099517            55.133333
3      trip-153671046011330457             0.983333
4      trip-153671052974046625             5.666667
…                           …                  …
14812  trip-153861095625827784             1.366667
14813  trip-153861104386292051             0.350000
14814  trip-153861106442901555             4.683333
14815  trip-153861115439069069             4.300000
14816  trip-153861118270144424             4.566667

[14817 rows x 2 columns]
```

[51]:
```
osrm_time = data.groupby(["trip_uuid",
              "start_scan_to_end_scan"])["osrm_time"].max().reset_index().
 ↪groupby("trip_uuid")["osrm_time"].sum().reset_index()
osrm_time
```

[51]:
```
                     trip_uuid  osrm_time
0      trip-153671041653548748  12.383333
1      trip-153671042288605164   1.133333
2      trip-153671043369099517  29.016667
3      trip-153671046011330457   0.250000
4      trip-153671052974046625   1.950000
…                           …         …
14812  trip-153861095625827784   1.033333
14813  trip-153861104386292051   0.200000
14814  trip-153861106442901555   0.900000
14815  trip-153861115439069069   3.066667
14816  trip-153861118270144424   1.133333

[14817 rows x 2 columns]
```

[52]:
```
time_taken_btwn_odstart_and_od_end = data.
 ↪groupby("trip_uuid")["time_taken_btwn_odstart_and_od_end"].unique().
 ↪reset_index()
time_taken_btwn_odstart_and_od_end
```

[52]:
```
                     trip_uuid  \
0      trip-153671041653548748
1      trip-153671042288605164
2      trip-153671043369099517
3      trip-153671046011330457
4      trip-153671052974046625
…                          …
14812  trip-153861095625827784
14813  trip-153861104386292051
14814  trip-153861106442901555
14815  trip-153861115439069069
14816  trip-153861118270144424

                      time_taken_btwn_odstart_and_od_end
0                             [16.65842298, 21.0100736875]
1                 [2.0463247669444447, 0.9805397955555556]
2                 [51.662059856388886, 13.910648811388889]
3                                     [1.6749155866666667]
4      [2.5335485744444446, 1.3423885633333332, 8.096…
…                                                      …
14812                   [2.546464057777778, 1.7540180775]
14813                                 [1.0098420219444444]
14814                   [2.895179575833333, 4.1401515375]
14815  [1.7609491794444445, 0.7362400538888889, 1.035…
14816                  [1.1155594141666667, 4.7912334425]

[14817 rows x 2 columns]
```

[53]:
```
time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"] =␣
↪time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].
↪apply(sum)
time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"]
```

[53]:
```
0        37.668497
1         3.026865
2        65.572709
3         1.674916
4        11.972484
           …
14812     4.300482
14813     1.009842
14814     7.035331
14815     5.808548
14816     5.906793
Name: time_taken_btwn_odstart_and_od_end, Length: 14817, dtype: float64
```

[54]:
```
start_scan_to_end_scan = ((data.groupby("trip_uuid")["start_scan_to_end_scan"].
 ↪unique())).reset_index()
start_scan_to_end_scan
```

[54]:
```
                    trip_uuid  \
0      trip-153671041653548748
1      trip-153671042288605164
2      trip-153671043369099517
3      trip-153671046011330457
4      trip-153671052974046625
…                          …
14812  trip-153861095625827784
14813  trip-153861104386292051
14814  trip-153861106442901555
14815  trip-153861115439069069
14816  trip-153861118270144424

                                  start_scan_to_end_scan
0                                             [16.65, 21.0]
1                 [2.033333333333333, 0.9666666666666667]
2                                             [51.65, 13.9]
3                                   [1.6666666666666667]
4      [2.533333333333333, 1.3333333333333333, 8.0833…
…                                                      …
14812                           [2.533333333333333, 1.75]
14813                                               [1.0]
14814             [2.8833333333333333, 4.133333333333334]
14815  [1.75, 0.7333333333333333, 1.0333333333333334,…
14816                           [1.1, 4.783333333333333]

[14817 rows x 2 columns]
```

[55]:
```
start_scan_to_end_scan["start_scan_to_end_scan"] =␣
 ↪start_scan_to_end_scan["start_scan_to_end_scan"].apply(sum)
start_scan_to_end_scan["start_scan_to_end_scan"]
```

[55]:
```
0        37.650000
1         3.000000
2        65.550000
3         1.666667
4        11.950000
           …
14812     4.283333
14813     1.000000
14814     7.016667
14815     5.783333
14816     5.883333
```

```
Name: start_scan_to_end_scan, Length: 14817, dtype: float64
```

[56]:
```
osrm_distance = data.groupby(["trip_uuid",
              "start_scan_to_end_scan"])["osrm_distance"].max().reset_index().
 ↪groupby("trip_uuid")["osrm_distance"].sum().reset_index()

osrm_distance
```

[56]:

	trip_uuid	osrm_distance
0	trip-153671041653548748	991.3523
1	trip-153671042288605164	85.1110
2	trip-153671043369099517	2372.0852
3	trip-153671046011330457	19.6800
4	trip-153671052974046625	146.7918
…	…	…
14812	trip-153861095625827784	73.4630
14813	trip-153861104386292051	16.0882
14814	trip-153861106442901555	63.2841
14815	trip-153861115439069069	177.6635
14816	trip-153861118270144424	80.5787

[14817 rows x 2 columns]

[57]:
```
actual_distance_to_destination = data.groupby(["trip_uuid",
              "start_scan_to_end_scan"])["actual_distance_to_destination"].
 ↪max().reset_index().groupby("trip_uuid")["actual_distance_to_destination"].
 ↪sum().reset_index()

actual_distance_to_destination
```

[57]:

	trip_uuid	actual_distance_to_destination
0	trip-153671041653548748	824.732854
1	trip-153671042288605164	73.186911
2	trip-153671043369099517	1932.273969
3	trip-153671046011330457	17.175274
4	trip-153671052974046625	127.448500
…	…	…
14812	trip-153861095625827784	57.762332
14813	trip-153861104386292051	15.513784
14814	trip-153861106442901555	38.684839
14815	trip-153861115439069069	134.723836
14816	trip-153861118270144424	66.081533

[14817 rows x 2 columns]

[59]:
```
segment_osrm_distance = data[["trip_uuid",
```

```
                                "segment_osrm_distance"]].
 ↪groupby("trip_uuid")["segment_osrm_distance"].sum().reset_index()

segment_osrm_distance
```

[59]:

	trip_uuid	segment_osrm_distance
0	trip-153671041653548748	1320.4733
1	trip-153671042288605164	84.1894
2	trip-153671043369099517	2545.2678
3	trip-153671046011330457	19.8766
4	trip-153671052974046625	146.7919
…	…	…
14812	trip-153861095625827784	64.8551
14813	trip-153861104386292051	16.0883
14814	trip-153861106442901555	104.8866
14815	trip-153861115439069069	223.5324
14816	trip-153861118270144424	80.5787

[14817 rows x 2 columns]

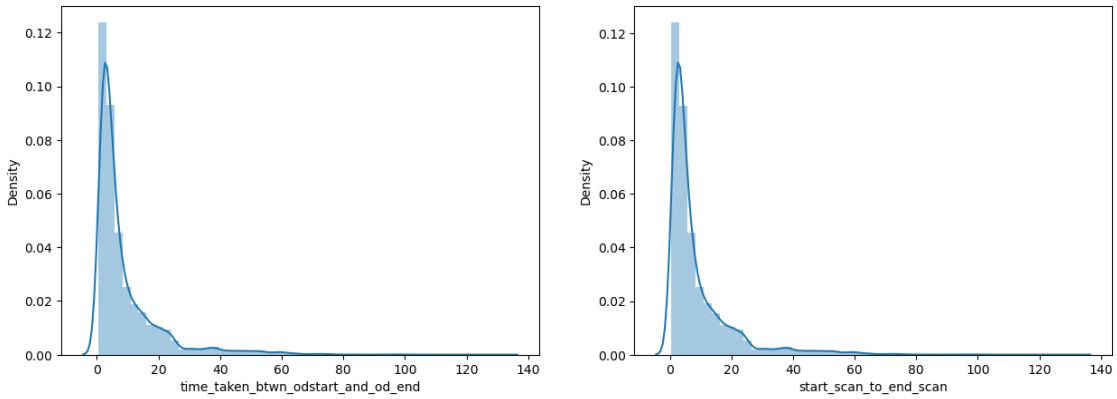
6 Hypothesis Test

6.0.1 Analysing TimeTaken Between OdStart and OdEnd time & StartScanToEndScan

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

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

[60]:
```
plt.figure(figsize=(15,5))
plt.subplot(121)
sns.
 ↪distplot((time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"]))
plt.subplot(122)
sns.distplot((start_scan_to_end_scan["start_scan_to_end_scan"]))

plt.show()
```



[62]:
```
ks_test, p_value = stats.
 ↪ks_2samp(time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"]
               ,start_scan_to_end_scan["start_scan_to_end_scan"])
```

[63]:
```
if p_value < 0.05:
    print("Reject Ho: The distribution are different.")
else :
    print("Fail to reject Ho: The distribution is same.")
```

```
Fail to reject Ho: The distribution is same.
```

[64]:
```
for i in range(5):
    print(stats.
 ↪ttest_ind((time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].
 ↪sample(3000))
                ,(start_scan_to_end_scan["start_scan_to_end_scan"].
 ↪sample(3000))))
```

```
Ttest_indResult(statistic=-2.1898143072655825, pvalue=0.02857609039464844)
Ttest_indResult(statistic=0.7436365550625867, pvalue=0.45712550973303223)
Ttest_indResult(statistic=0.7193981709953762, pvalue=0.47192365108776835)
Ttest_indResult(statistic=-0.3822681163017748, pvalue=0.7022760451887388)
Ttest_indResult(statistic=-0.31840545932365827, pvalue=0.7501884834741348)
```

[65]:
```
time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].
 ↪mean(),time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].
 ↪std()
```

[65]:
```
(8.861857235305113, 10.981665759990623)
```

[66]:
```
start_scan_to_end_scan["start_scan_to_end_scan"].
 ↪mean(),start_scan_to_end_scan["start_scan_to_end_scan"].std()
```

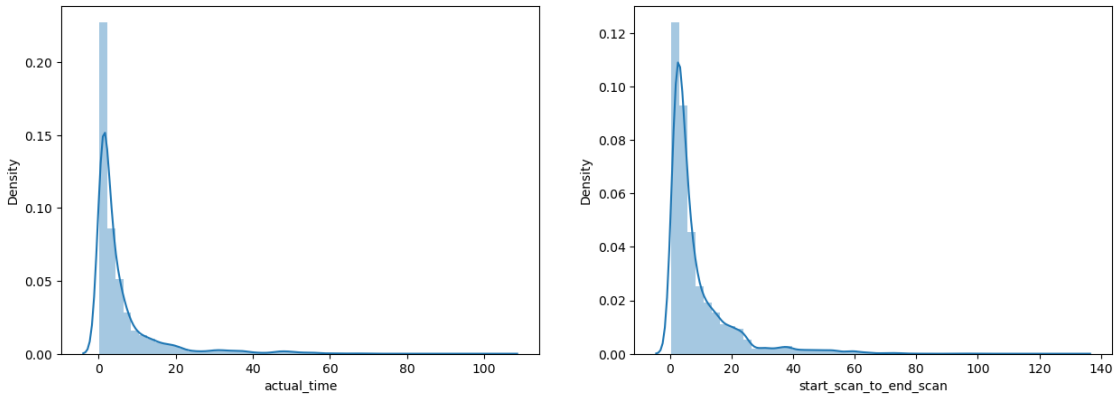
[66]:
```
(8.835777597804324, 10.97628639143973)
```

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

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

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

[67]:
```
plt.figure(figsize=(15,5))
plt.subplot(121)
sns.distplot((actual_time["actual_time"]))
plt.subplot(122)
sns.distplot((start_scan_to_end_scan["start_scan_to_end_scan"]))

plt.show()
```



[68]:
```
stats.
 ↪ks_2samp(actual_time["actual_time"],start_scan_to_end_scan["start_scan_to_end_scan"])
```

[68]:
```
KstestResult(statistic=0.27387460349598436, pvalue=0.0,
statistic_location=1.8499999999999999, statistic_sign=1)
```

[69]:
```
for i in range(7):
    print(stats.ttest_ind((actual_time["actual_time"].sample(3000))
                ,(start_scan_to_end_scan["start_scan_to_end_scan"].
 ↪sample(3000)),alternative="less"))
```

```
Ttest_indResult(statistic=-10.636025114359327, pvalue=1.7314291485139564e-26)
Ttest_indResult(statistic=-12.05470607102136, pvalue=2.2040624413304303e-33)
Ttest_indResult(statistic=-9.793040937949158, pvalue=8.882340694243722e-23)
Ttest_indResult(statistic=-10.050824468449582, pvalue=6.99494321140646e-24)
Ttest_indResult(statistic=-10.934783052396664, pvalue=7.145539555328463e-28)
Ttest_indResult(statistic=-10.443059994033343, pvalue=1.297972859685586e-25)
Ttest_indResult(statistic=-10.528980490964472, pvalue=5.31610350507997e-26)
```

[70]:
```
actual_time["actual_time"].mean(),actual_time["actual_time"].std()
```

[70]:
```
(5.945176711435065, 9.35554782297388)
```

[71]:
```
start_scan_to_end_scan["start_scan_to_end_scan"].
↪mean(),start_scan_to_end_scan["start_scan_to_end_scan"].std()
```

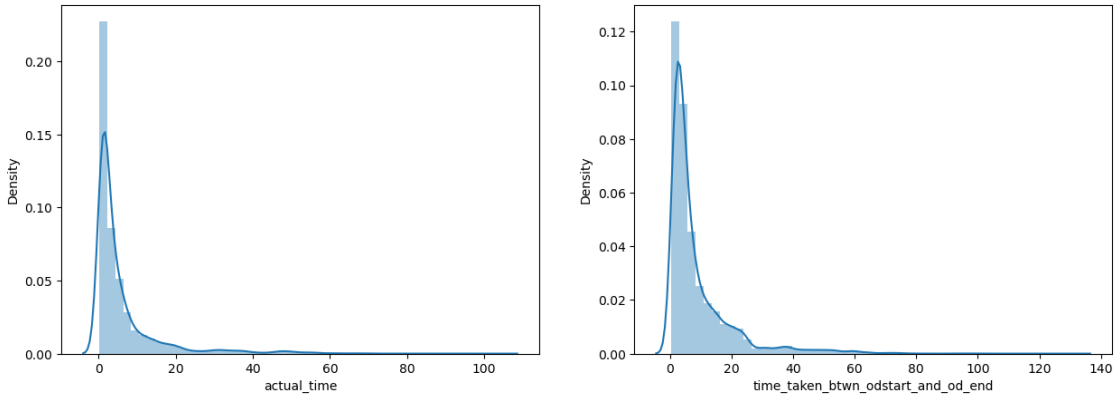
[71]:
```
(8.835777597804324, 10.97628639143973)
```

6.0.3 Analysing Actual Time & TimeTaken between start and end trip time.

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

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

[72]:
```
plt.figure(figsize=(15,5))
plt.subplot(121)
sns.distplot((actual_time["actual_time"]))
plt.subplot(122)
sns.
↪distplot((time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"]))

plt.show()
```



[73]:
```
stats.
↪ks_2samp(actual_time["actual_time"],time_taken_btwn_odstart_and_od_end["time_taken_btwn_ods
```

[73]:
```
KstestResult(statistic=0.2765067152594992, pvalue=0.0,
statistic_location=1.8333333333333335, statistic_sign=1)
```

[74]:
```
for i in range(5):
    print(stats.ttest_ind((actual_time["actual_time"].sample(1000))
```

```
                                                                    ␣
↪,(time_taken_btwn_odstart_and_od_end["time_taken_btwn_odstart_and_od_end"].
↪sample(1000))))
```

```
Ttest_indResult(statistic=-8.517757266213723, pvalue=3.150009184926608e-17)
Ttest_indResult(statistic=-4.6455345054973805, pvalue=3.6130467903119877e-06)
Ttest_indResult(statistic=-6.647714823945244, pvalue=3.828579041291592e-11)
Ttest_indResult(statistic=-6.1846568852185095, pvalue=7.527148027696814e-10)
Ttest_indResult(statistic=-6.243492254290337, pvalue=5.212130873569678e-10)
```

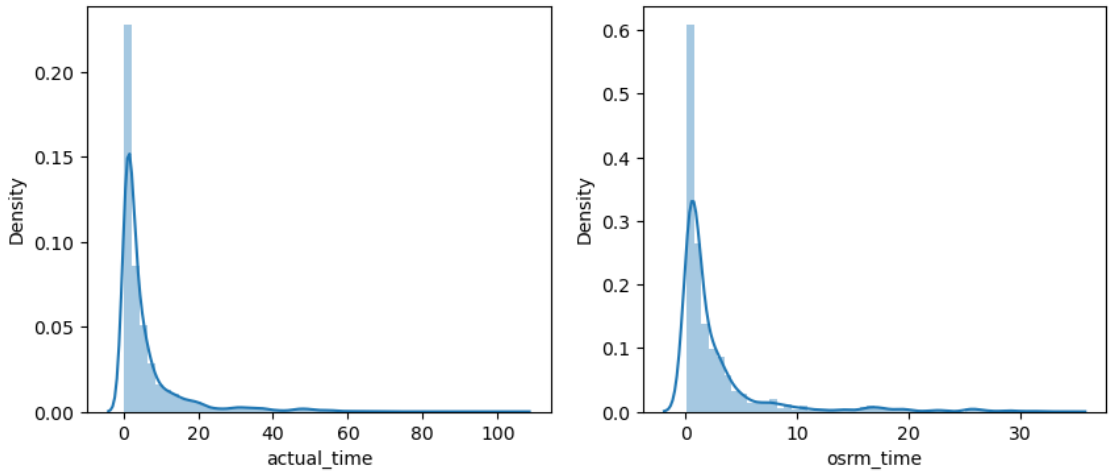
from above kstest of distribution and two sample ttest , we can conclude that population mean Actual time taken to complete delivery and population mean time_taken_btwn_od_start_and_od_end are also not same.

6.0.4 Analysing Actual Time taken to complete delivery from source to destination hub & OSRM measured time

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

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

[75]:
```
plt.figure(figsize=(10,4))
plt.subplot(121)
sns.distplot(((actual_time["actual_time"])))
plt.subplot(122)
sns.distplot(((osrm_time["osrm_time"])))

plt.show()
```



[76]:
```
stats.ks_2samp(actual_time["actual_time"],
               osrm_time["osrm_time"])
```

[76]:
```
KstestResult(statistic=0.2945265573327934, pvalue=0.0,
statistic_location=0.6833333333333333, statistic_sign=-1)
```

[77]:
```
for i in range(5):
    print(stats.ttest_ind(actual_time["actual_time"].sample(5000),
                osrm_time["osrm_time"].sample(5000),alternative='greater'))
```

```
Ttest_indResult(statistic=23.041808906554632, pvalue=8.256294836384888e-115)
Ttest_indResult(statistic=21.635719578428485, pvalue=8.609350969474735e-102)
Ttest_indResult(statistic=22.697301070547446, pvalue=1.4947371049381022e-111)
Ttest_indResult(statistic=21.593704725773073, pvalue=2.0538368454687474e-101)
Ttest_indResult(statistic=21.69981165118125, pvalue=2.278670570405785e-102)
```

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

Actual time is higher than the osrm estimated time for delivery.

[78]:
```
actual_time["actual_time"].mean(),actual_time["actual_time"].std()
```

[78]:
```
(5.945176711435065, 9.35554782297388)
```

[80]:
```
osrm_time["osrm_time"].mean(),osrm_time["osrm_time"].std()
```

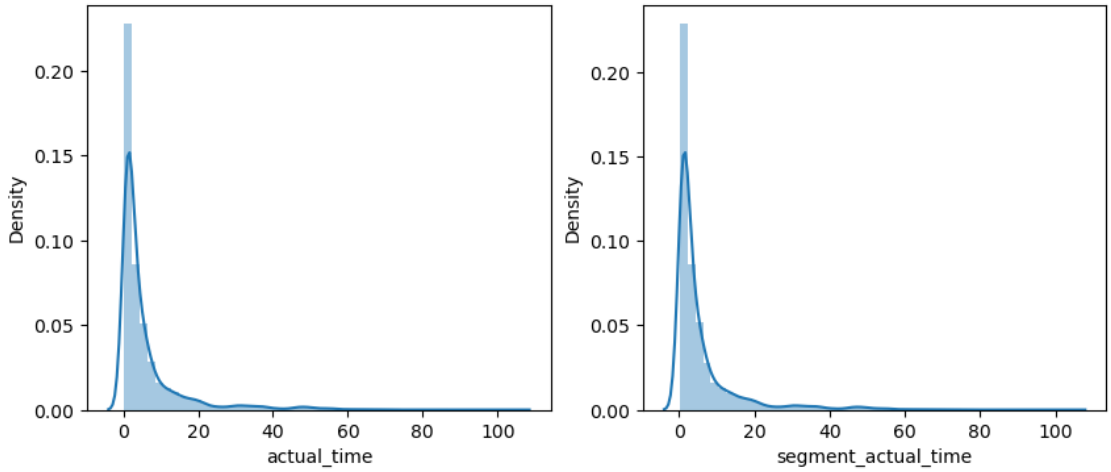
[80]:
```
(2.6973138962003107, 4.537654251845703)
```

6.0.5 Analysing Actual Time taken to complete delivery from source to destination hub & Segment Actual Time

H0: Actual time = segment actual time

Ha: Actual time != segment actual time

[81]:
```
plt.figure(figsize=(10,4))
plt.subplot(121)
sns.distplot(((actual_time["actual_time"])))
plt.subplot(122)
sns.distplot(((segment_actual_time["segment_actual_time"])))

plt.show()
```



[82]:
```
for i in range(7):
    print(stats.ttest_ind((actual_time["actual_time"].sample(3000)),
                (segment_actual_time["segment_actual_time"].sample(3000))))
```

```
Ttest_indResult(statistic=1.1490261350446236, pvalue=0.25059102148250967)
Ttest_indResult(statistic=0.19315950737842924, pvalue=0.8468405900570704)
Ttest_indResult(statistic=0.5248442433358365, pvalue=0.5997108696152572)
Ttest_indResult(statistic=0.10776142240828243, pvalue=0.9141885493859879)
Ttest_indResult(statistic=0.9458507722142563, pvalue=0.3442628060575105)
Ttest_indResult(statistic=0.5307530901855176, pvalue=0.5956095412702898)
Ttest_indResult(statistic=0.6323657762695039, pvalue=0.5271719956433005)
```

[83]:
```
actual_time["actual_time"].mean(),actual_time["actual_time"].std()
```

[83]:
```
(5.945176711435065, 9.35554782297388)
```

[84]:
```
segment_actual_time["segment_actual_time"].
↪mean(),segment_actual_time["segment_actual_time"].std()
```

[84]:
```
(5.8982047647971925, 9.270799413152762)
```

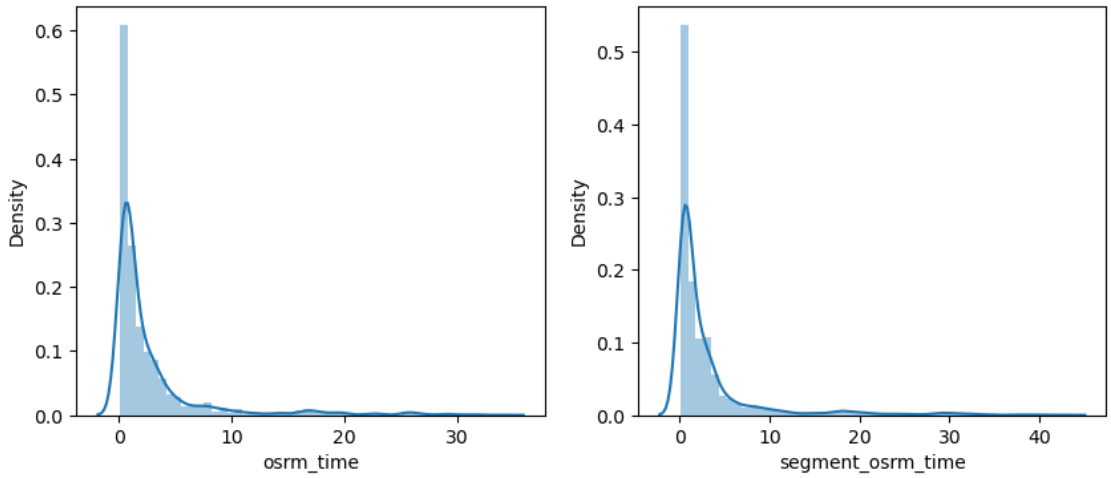
6.0.6 Analysing osrm Time & segment-osrm-time

Ho: segment actual time <= OSRM time

Ha: segment actual time > OSRM time

[85]:
```
plt.figure(figsize=(10,4))
plt.subplot(121)
sns.distplot(((osrm_time["osrm_time"])))
plt.subplot(122)
sns.distplot(((segment_osrm_time["segment_osrm_time"])))
```

```
plt.show()
```



[86]:
```
for i in range(7):
    print(stats.ttest_ind((osrm_time["osrm_time"].sample(3000)),
                (segment_osrm_time["segment_osrm_time"].
 ↪sample(3000)),alternative ="less"))
```

```
Ttest_indResult(statistic=-3.6758135209219223, pvalue=0.0001195818626731392)
Ttest_indResult(statistic=-3.3612655823081106, pvalue=0.0003903556263434343144)
Ttest_indResult(statistic=-2.824590656965777, pvalue=0.0023748666777830625)
Ttest_indResult(statistic=-0.6315732133658719, pvalue=0.26384493158307787)
Ttest_indResult(statistic=-3.017132784033036, pvalue=0.0012812439192925874)
Ttest_indResult(statistic=-3.9075003833008255, pvalue=4.7141077740903e-05)
Ttest_indResult(statistic=-3.694930458026216, pvalue=0.0001109530792748311)
```

From ttest , we can conclude that

1. Average of osrm Time & segment-osrm-time for population is not same.
2. Population Mean osrm time is less than Population Mean segment osrm time

[87]:
```
osrm_time["osrm_time"].mean(),osrm_time["osrm_time"].std()
```

[87]: (2.6973138962003107, 4.537654251845703)

[88]:
```
segment_osrm_time["segment_osrm_time"].
 ↪mean(),segment_osrm_time["segment_osrm_time"].std()
```

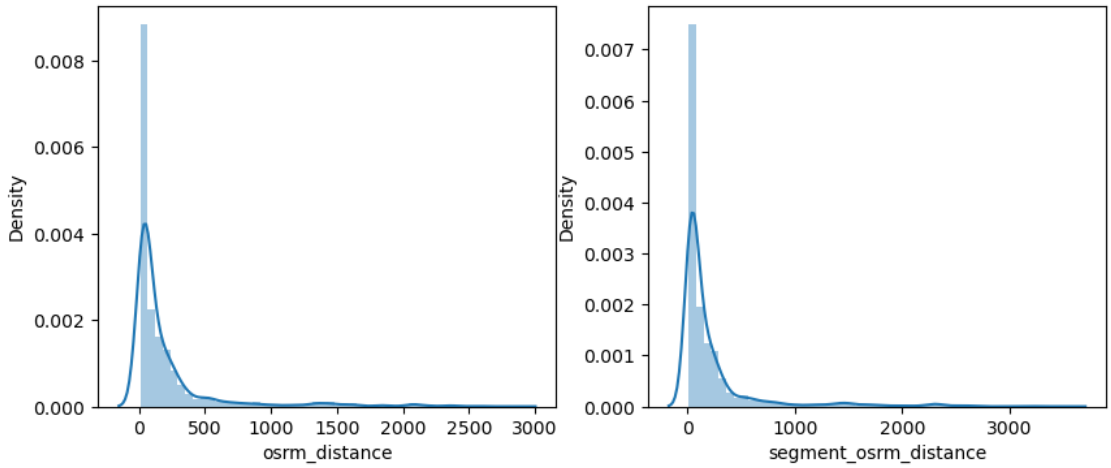
[88]: (3.0158297901059594, 5.242367441693007)

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

H0 : Segment OSRM distnace <= OSRM distnace

Ha : Segment OSRM distnace > OSRM distnace

[89]:
```
plt.figure(figsize=(10,4))
plt.subplot(121)
sns.distplot(((osrm_distance["osrm_distance"])))
plt.subplot(122)
sns.distplot(((segment_osrm_distance["segment_osrm_distance"])))

plt.show()
```



[90]:
```
stats.
↪ks_2samp(osrm_distance["osrm_distance"],segment_osrm_distance["segment_osrm_distance"])
```

[90]:
```
KstestResult(statistic=0.03948167645272321, pvalue=1.8042208791084262e-10,
statistic_location=50.2941, statistic_sign=1)
```

[91]:
```
for i in range(7):
    print(stats.ttest_ind(osrm_distance["osrm_distance"].sample(5000),
                segment_osrm_distance["segment_osrm_distance"].
↪sample(5000),alternative="less"))
```

```
Ttest_indResult(statistic=-2.5652973134123283, pvalue=0.00516159784798941)
Ttest_indResult(statistic=-2.9209747937835777, pvalue=0.0017485913303405497)
Ttest_indResult(statistic=-2.603616446902811, pvalue=0.004619112736780279)
Ttest_indResult(statistic=-2.975639191932269, pvalue=0.0014653902520709611)
Ttest_indResult(statistic=-0.895385090506525, pvalue=0.18530141183244975)
Ttest_indResult(statistic=-3.6921444336088434, pvalue=0.0001117773232663314)
Ttest_indResult(statistic=-1.8152761587128723, pvalue=0.034755713900821036)
```

[92]:
```
osrm_distance["osrm_distance"].mean(),osrm_distance["osrm_distance"].std()
```

[92]:
```
(204.83672531551593, 370.74927471335496)
```

[93]:
```
segment_osrm_distance["segment_osrm_distance"].
↪mean(),segment_osrm_distance["segment_osrm_distance"].std()
```

[93]:
```
(223.20116128771005, 416.6283742907418)
```

From KS test , we can conclude the distributions of segment osrm distance and osrm distnace are not same

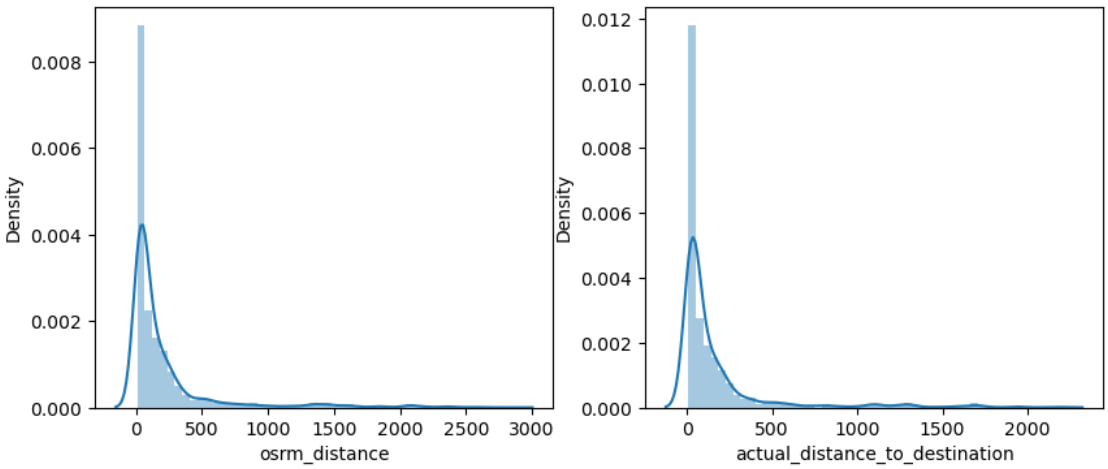
From two sample one sided ttest, we can conclude: Average of osrm distance for population is less than average of segment osrm distnace

6.0.8 Analysing and Visulizing OSRM Estimated distance and Actual Distance between source and destination warehouse

H0 : Mean OSRM distance <= Mean Actual distnace

Ha : Mean OSRM distance > Mean Actual distnace

[94]:
```
plt.figure(figsize=(10,4))
plt.subplot(121)
sns.distplot(((osrm_distance["osrm_distance"])))
plt.subplot(122)
sns.
↪distplot(((actual_distance_to_destination["actual_distance_to_destination"])))

plt.show()
```



[95]:
```
stats.
↪ks_2samp(osrm_distance["osrm_distance"],actual_distance_to_destination["actual_distance_to_
```

[95]:
```
KstestResult(statistic=0.11837753931295136, pvalue=6.578385372142345e-91,
statistic_location=25.247485296255537, statistic_sign=-1)
```

[96]:
```
for i in range(5):
    print(stats.ttest_ind(osrm_distance["osrm_distance"].sample(5000),
                actual_distance_to_destination["actual_distance_to_destination"].
 ↪sample(5000),alternative="greater"))
```

```
Ttest_indResult(statistic=6.472018425278162, pvalue=5.0618108479633035e-11)
Ttest_indResult(statistic=5.784003899103792, pvalue=3.756634043410597e-09)
Ttest_indResult(statistic=5.361646054695082, pvalue=4.2152411344577114e-08)
Ttest_indResult(statistic=6.110016721307429, pvalue=5.166748908501008e-10)
Ttest_indResult(statistic=6.318616700983314, pvalue=1.3757975176144214e-10)
```

[97]:
```
osrm_distance["osrm_distance"].mean(),osrm_distance["osrm_distance"].std()
```

[97]:
```
(204.83672531551593, 370.74927471335496)
```

[98]:
```
actual_distance_to_destination["actual_distance_to_destination"].
 ↪mean(),actual_distance_to_destination["actual_distance_to_destination"].std()
```

[98]:
```
(164.47332174544243, 305.5408288910492)
```

7 Merging

[99]:
```
distances = segment_osrm_distance.merge(actual_distance_to_destination.
 ↪merge(osrm_distance,
                                                                          ␣
 ↪on="trip_uuid"),
                                                                         ␣
 ↪on="trip_uuid")
```

[100]:
```
distances
```

[100]:

	trip_uuid	segment_osrm_distance	\
0	trip-153671041653548748	1320.4733	
1	trip-153671042288605164	84.1894	
2	trip-153671043369099517	2545.2678	
3	trip-153671046011330457	19.8766	
4	trip-153671052974046625	146.7919	
…	…	…	
14812	trip-153861095625827784	64.8551	
14813	trip-153861104386292051	16.0883	
14814	trip-153861106442901555	104.8866	
14815	trip-153861115439069069	223.5324	
14816	trip-153861118270144424	80.5787	

```
       actual_distance_to_destination  osrm_distance
0                           824.732854       991.3523
1                            73.186911        85.1110
2                          1932.273969      2372.0852
3                            17.175274        19.6800
4                           127.448500       146.7918
…                                  …              …
14812                        57.762332        73.4630
14813                        15.513784        16.0882
14814                        38.684839        63.2841
14815                       134.723836       177.6635
14816                        66.081533        80.5787

[14817 rows x 4 columns]
```

[101]:
```
time = segment_osrm_time.merge(osrm_time.merge(segment_actual_time.
↪merge(actual_time.merge(time_taken_btwn_odstart_and_od_end.
↪merge(start_scan_to_end_scan,
                                        on="trip_uuid",
                                       ␣
↪),on="trip_uuid"),on="trip_uuid"),on="trip_uuid"),on="trip_uuid")

time
```

[101]:
```
                    trip_uuid  segment_osrm_time  osrm_time  \
0      trip-153671041653548748          16.800000  12.383333
1      trip-153671042288605164           1.083333   1.133333
2      trip-153671043369099517          32.350000  29.016667
3      trip-153671046011330457           0.266667   0.250000
4      trip-153671052974046625           1.916667   1.950000
…                          …                 …          …
14812  trip-153861095625827784           1.033333   1.033333
14813  trip-153861104386292051           0.183333   0.200000
14814  trip-153861106442901555           1.466667   0.900000
14815  trip-153861115439069069           3.683333   3.066667
14816  trip-153861118270144424           1.116667   1.133333

       segment_actual_time  actual_time  time_taken_btwn_odstart_and_od_end  \
0                25.800000    26.033333                           37.668497
1                 2.350000     2.383333                            3.026865
2                55.133333    55.783333                           65.572709
3                 0.983333     0.983333                            1.674916
4                 5.666667     5.683333                           11.972484
…                      …            …                                   …
14812             1.366667     1.383333                            4.300482
14813             0.350000     0.350000                            1.009842
14814             4.683333     4.700000                            7.035331
```

```
14815             4.300000     4.400000                            5.808548
14816             4.566667     4.583333                            5.906793

       start_scan_to_end_scan
0                   37.650000
1                    3.000000
2                   65.550000
3                    1.666667
4                   11.950000
…                         …
14812                4.283333
14813                1.000000
14814                7.016667
14815                5.783333
14816                5.883333

[14817 rows x 7 columns]
```

[102]:
```
Merge1 = time.merge(distances,on="trip_uuid",
                                    )
Merge1
```

[102]:
```
                    trip_uuid  segment_osrm_time  osrm_time  \
0      trip-153671041653548748           16.800000  12.383333
1      trip-153671042288605164            1.083333   1.133333
2      trip-153671043369099517           32.350000  29.016667
3      trip-153671046011330457            0.266667   0.250000
4      trip-153671052974046625            1.916667   1.950000
…                          …                  …         …
14812  trip-153861095625827784            1.033333   1.033333
14813  trip-153861104386292051            0.183333   0.200000
14814  trip-153861106442901555            1.466667   0.900000
14815  trip-153861115439069069            3.683333   3.066667
14816  trip-153861118270144424            1.116667   1.133333

       segment_actual_time  actual_time  time_taken_btwn_odstart_and_od_end  \
0                25.800000    26.033333                           37.668497
1                 2.350000     2.383333                            3.026865
2                55.133333    55.783333                           65.572709
3                 0.983333     0.983333                            1.674916
4                 5.666667     5.683333                           11.972484
…                      …            …                                   …
14812             1.366667     1.383333                            4.300482
14813             0.350000     0.350000                            1.009842
14814             4.683333     4.700000                            7.035331
14815             4.300000     4.400000                            5.808548
14816             4.566667     4.583333                            5.906793
```

```
       start_scan_to_end_scan  segment_osrm_distance  \
0                   37.650000             1320.4733
1                    3.000000               84.1894
2                   65.550000             2545.2678
3                    1.666667               19.8766
4                   11.950000              146.7919
…                         …                     …
14812                4.283333               64.8551
14813                1.000000               16.0883
14814                7.016667              104.8866
14815                5.783333              223.5324
14816                5.883333               80.5787

       actual_distance_to_destination  osrm_distance
0                          824.732854       991.3523
1                           73.186911        85.1110
2                         1932.273969      2372.0852
3                           17.175274        19.6800
4                          127.448500       146.7918
…                                 …              …
14812                       57.762332        73.4630
14813                       15.513784        16.0882
14814                       38.684839        63.2841
14815                      134.723836       177.6635
14816                       66.081533        80.5787

[14817 rows x 10 columns]
```

8 Merging Location details and route_type and Numerical data on TripID

[103]:
```
city = data.groupby("trip_uuid")[["source_city",
                                  "destination_city"]].aggregate({
        "source_city":pd.unique,
    "destination_city":pd.unique,
})

state = data.groupby("trip_uuid")[["source_state",
                                   "destination_state"]].aggregate({
        "source_state":pd.unique,
    "destination_state":pd.unique,
})

city_state = data.groupby("trip_uuid")[["source_city_state",
                                        "destination_city_state"]].aggregate({
```

```
        "source_city_state":pd.unique,
    "destination_city_state":pd.unique,
})

locations = city.merge(city_state.merge(state,on="trip_uuid"
                            ,how="outer"),
           on="trip_uuid",
           how="outer")
```

[104]:
```
route_type = data.groupby("trip_uuid")["route_type"].unique().reset_index()

Merged = route_type.merge(locations.merge(Merge1,on="trip_uuid",
           how="outer"),
                 on="trip_uuid",
           how="outer"
                )
```

[105]:
```
trip_records = Merged.copy()
```

[106]:
```
trip_records["route_type"] = trip_records["route_type"].apply(lambda x:x[0])
route_to_merge = data.groupby("trip_uuid")["route_schedule_uuid"].unique().
 ↪reset_index()
trip_records = trip_records.merge(route_to_merge,on="trip_uuid",how="outer")
trip_records["route_schedule_uuid"] = trip_records["route_schedule_uuid"].
 ↪apply(lambda x:x[0])
trip_records
```

[106]:

	trip_uuid	route_type	source_city
0	trip-153671041653548748	FTL	[Bhopal, Kanpur]
1	trip-153671042288605164	Carting	[Tumkur, Doddablpur]
2	trip-153671043369099517	FTL	[Bengaluru, Gurgaon]
3	trip-153671046011330457	Carting	[Mumbai]
4	trip-153671052974046625	FTL	[Bellary, Hospet, Sandur]
…	…	…	…
14812	trip-153861095625827784	Carting	
14813	trip-153861104386292051	Carting	
14814	trip-153861106442901555	Carting	
14815	trip-153861115439069069	Carting	
14816	trip-153861118270144424	FTL	

```
14812                                        [Chandigarh]
14813                                               [FBD]
14814                                           [Kanpur]
14815  [Tirunelveli, Eral, Tirchchndr, Thisayanvilai,…
14816                                    [Hospet, Sandur]

                                      destination_city  \
0                                     [Kanpur, Gurgaon]
1                                [Doddablpur, Chikblapur]
2                                   [Gurgaon, Chandigarh]
3                                               [Mumbai]
4                               [Hospet, Sandur, Bellary]
…                                                     …
14812                               [Zirakpur, Chandigarh]
14813                                        [Faridabad]
14814                                           [Kanpur]
14815  [Eral, Tirchchndr, Thisayanvilai, Peikulam, Ti…
14816                                    [Sandur, Bellary]

                                    source_city_state  \
0          [Bhopal Madhya Pradesh, Kanpur Uttar Pradesh]
1               [Tumkur Karnataka, Doddablpur Karnataka]
2                 [Bengaluru Karnataka, Gurgaon Haryana]
3                               [Mumbai Hub Maharashtra]
4      [Bellary Karnataka, Hospet Karnataka, Sandur K…
…                                                     …
14812         [Chandigarh Punjab, Chandigarh Chandigarh]
14813                                      [FBD Haryana]
14814                               [Kanpur Uttar Pradesh]
14815  [Tirunelveli Tamil Nadu, Eral Tamil Nadu, Tirc…
14816                [Hospet Karnataka, Sandur Karnataka]

                                 destination_city_state  \
0                [Kanpur Uttar Pradesh, Gurgaon Haryana]
1            [Doddablpur Karnataka, Chikblapur Karnataka]
2                    [Gurgaon Haryana, Chandigarh Punjab]
3                                  [Mumbai Maharashtra]
4      [Hospet Karnataka, Sandur Karnataka, Bellary K…
…                                                     …
14812                [Zirakpur Punjab, Chandigarh Punjab]
14813                                 [Faridabad Haryana]
14814                               [Kanpur Uttar Pradesh]
14815  [Eral Tamil Nadu, Tirchchndr Tamil Nadu, Thisa…
14816                [Sandur Karnataka, Bellary Karnataka]

                         source_state         destination_state  \
0      [Madhya Pradesh, Uttar Pradesh]  [Uttar Pradesh, Haryana]
```

```
1                          [Karnataka]              [Karnataka]
2                 [Karnataka, Haryana]         [Haryana, Punjab]
3                    [Hub Maharashtra]             [Maharashtra]
4                          [Karnataka]              [Karnataka]
…                                  …                        …
14812             [Punjab, Chandigarh]                [Punjab]
14813                        [Haryana]               [Haryana]
14814                    [Uttar Pradesh]          [Uttar Pradesh]
14815                       [Tamil Nadu]             [Tamil Nadu]
14816                       [Karnataka]              [Karnataka]

       segment_osrm_time  osrm_time  segment_actual_time  actual_time  \
0              16.800000  12.383333            25.800000    26.033333
1               1.083333   1.133333             2.350000     2.383333
2              32.350000  29.016667            55.133333    55.783333
3               0.266667   0.250000             0.983333     0.983333
4               1.916667   1.950000             5.666667     5.683333
…                    …          …                    …            …
14812           1.033333   1.033333             1.366667     1.383333
14813           0.183333   0.200000             0.350000     0.350000
14814           1.466667   0.900000             4.683333     4.700000
14815           3.683333   3.066667             4.300000     4.400000
14816           1.116667   1.133333             4.566667     4.583333

       time_taken_btwn_odstart_and_od_end  start_scan_to_end_scan  \
0                                37.668497               37.650000
1                                 3.026865                3.000000
2                                65.572709               65.550000
3                                 1.674916                1.666667
4                                11.972484               11.950000
…                                      …                       …
14812                             4.300482                4.283333
14813                             1.009842                1.000000
14814                             7.035331                7.016667
14815                             5.808548                5.783333
14816                             5.906793                5.883333

       segment_osrm_distance  actual_distance_to_destination  osrm_distance  \
0                  1320.4733                       824.732854       991.3523
1                    84.1894                        73.186911        85.1110
2                  2545.2678                      1932.273969      2372.0852
3                    19.8766                        17.175274        19.6800
4                   146.7919                       127.448500       146.7918
…                        …                              …              …
14812                64.8551                        57.762332        73.4630
14813                16.0883                        15.513784        16.0882
14814               104.8866                        38.684839        63.2841
```

```
14815                223.5324                      134.723836       177.6635
14816                 80.5787                       66.081533        80.5787

                                        route_schedule_uuid
0      thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…
1      thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…
2      thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e…
3      thanos::sroute:f0176492-a679-4597-8332-bbd1c7f…
4      thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134…
…                                                    …
14812  thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14…
14813  thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769…
14814  thanos::sroute:5609c268-e436-4e0a-8180-3db4a74…
14815  thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a…
14816  thanos::sroute:412fea14-6d1f-4222-8a5f-a517042…

[14817 rows x 18 columns]
```

[107]:
```
trip_records.isna().sum()
```

[107]:
```
trip_uuid                           0
route_type                          0
source_city                         0
destination_city                    0
source_city_state                   0
destination_city_state              0
source_state                        0
destination_state                   0
segment_osrm_time                   0
osrm_time                           0
segment_actual_time                 0
actual_time                         0
time_taken_btwn_odstart_and_od_end  0
start_scan_to_end_scan              0
segment_osrm_distance               0
actual_distance_to_destination      0
osrm_distance                       0
route_schedule_uuid                 0
dtype: int64
```

8.1 Unnesting Data

[108]:
```
trip_records["source_city"] = trip_records["source_city"].astype("str").str.
 ↪strip("[]").str.replace("'","")
trip_records["destination_city"] = trip_records["destination_city"].
 ↪astype("str").str.strip("[]").str.replace("'","")
```

```
trip_records["source_city_state"] = trip_records["source_city_state"].
 ↪astype("str").str.strip("[]").str.replace("'","")
trip_records["destination_city_state"] = trip_records["destination_city_state"].
 ↪astype("str").str.strip("[]").str.replace("'","")

trip_records["source_state"] = trip_records["source_state"].astype("str").str.
 ↪strip("[]").str.replace("'","")
trip_records["destination_state"] = trip_records["destination_state"].
 ↪astype("str").str.strip("[]").str.replace("'","")
```

[109]:
```
trip_records.corr()
```

[109]:

	segment_osrm_time	osrm_time
segment_osrm_time	1.000000	0.993508
osrm_time	0.993508	1.000000
segment_actual_time	0.953039	0.957747
actual_time	0.953800	0.958613
time_taken_btwn_odstart_and_od_end	0.918447	0.926280
start_scan_to_end_scan	0.918493	0.926469
segment_osrm_distance	0.996092	0.991848
actual_distance_to_destination	0.987627	0.993556
osrm_distance	0.992050	0.997610

	segment_actual_time	actual_time
segment_osrm_time	0.953039	0.953800
osrm_time	0.957747	0.958613
segment_actual_time	1.000000	0.999920
actual_time	0.999920	1.000000
time_taken_btwn_odstart_and_od_end	0.961096	0.960958
start_scan_to_end_scan	0.961107	0.961163
segment_osrm_distance	0.956106	0.956949
actual_distance_to_destination	0.953048	0.954082
osrm_distance	0.958341	0.959290

	time_taken_btwn_odstart_and_od_end
segment_osrm_time	0.918447
osrm_time	0.926280
segment_actual_time	0.961096
actual_time	0.960958
time_taken_btwn_odstart_and_od_end	1.000000
start_scan_to_end_scan	0.999860
segment_osrm_distance	0.919156
actual_distance_to_destination	0.918373
osrm_distance	0.924093

	start_scan_to_end_scan
segment_osrm_time	0.918493

osrm_time	0.926469
segment_actual_time	0.961107
actual_time	0.961163
time_taken_btwn_odstart_and_od_end	0.999860
start_scan_to_end_scan	1.000000
segment_osrm_distance	0.919288
actual_distance_to_destination	0.918671
osrm_distance	0.924368

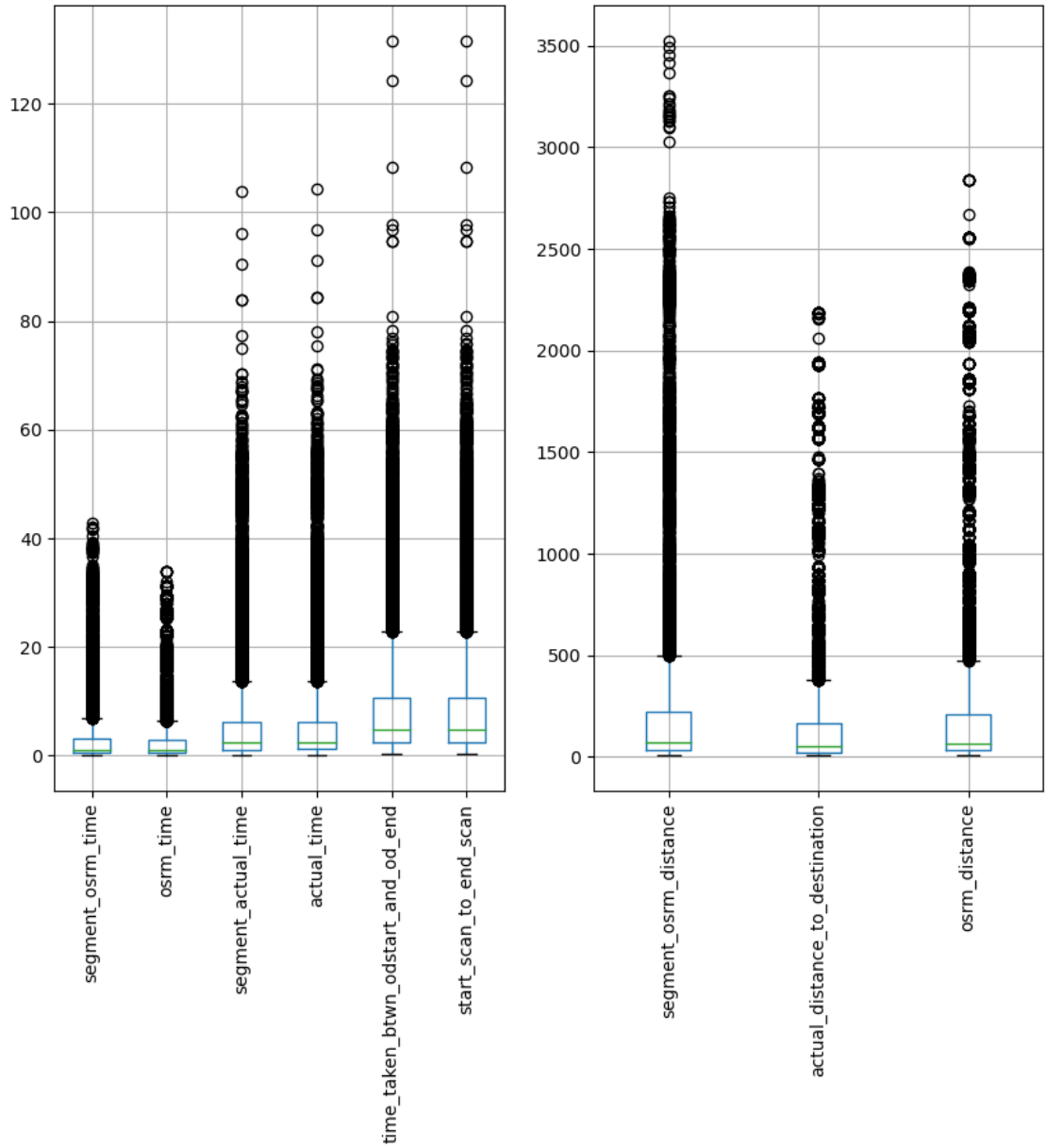
	segment_osrm_distance \
segment_osrm_time	0.996092
osrm_time	0.991848
segment_actual_time	0.956106
actual_time	0.956949
time_taken_btwn_odstart_and_od_end	0.919156
start_scan_to_end_scan	0.919288
segment_osrm_distance	1.000000
actual_distance_to_destination	0.993207
osrm_distance	0.994921

	actual_distance_to_destination \
segment_osrm_time	0.987627
osrm_time	0.993556
segment_actual_time	0.953048
actual_time	0.954082
time_taken_btwn_odstart_and_od_end	0.918373
start_scan_to_end_scan	0.918671
segment_osrm_distance	0.993207
actual_distance_to_destination	1.000000
osrm_distance	0.997273

	osrm_distance
segment_osrm_time	0.992050
osrm_time	0.997610
segment_actual_time	0.958341
actual_time	0.959290
time_taken_btwn_odstart_and_od_end	0.924093
start_scan_to_end_scan	0.924368
segment_osrm_distance	0.994921
actual_distance_to_destination	0.997273
osrm_distance	1.000000

9 Detecting Outliers

[110]:
```
plt.figure(figsize = (10,8))
plt.subplot(121)
trip_records[['segment_osrm_time', 'osrm_time',
       'segment_actual_time', 'actual_time',
       'time_taken_btwn_odstart_and_od_end', 'start_scan_to_end_scan']].
 ↪boxplot()
plt.xticks(rotation =90)
plt.subplot(122)
trip_records[['segment_osrm_distance', 'actual_distance_to_destination',
       'osrm_distance']].boxplot()
plt.xticks(rotation =90)
plt.show()
```



[111]:
```
outlier_treatment  = trip_records.copy()
```

[112]:
```
outlier_treatment_num = outlier_treatment[['segment_osrm_time', 'osrm_time',
       'segment_actual_time', 'actual_time',
       'time_taken_btwn_odstart_and_od_end', 'start_scan_to_end_scan',
        'segment_osrm_distance', 'actual_distance_to_destination',
        'osrm_distance']]
```

[113]:
```
trip_records_without_outliers = trip_records.loc[outlier_treatment_num[(np.
↪abs(stats.zscore(outlier_treatment_num)) < 3).all(axis=1)].index]
trip_records_without_outliers
```

[113]:

```
                    trip_uuid route_type  \
0      trip-153671041653548748        FTL
1      trip-153671042288605164    Carting
3      trip-153671046011330457    Carting
4      trip-153671052974046625        FTL
5      trip-153671055416136166    Carting
…                          …          …
14812  trip-153861095625827784    Carting
14813  trip-153861104386292051    Carting
14814  trip-153861106442901555    Carting
14815  trip-153861115439069069    Carting
14816  trip-153861118270144424        FTL

                                          source_city  \
0                                        Bhopal Kanpur
1                                    Tumkur Doddablpur
3                                               Mumbai
4                                 Bellary Hospet Sandur
5                                             Chennai
…                                                 …
14812                                      Chandigarh
14813                                             FBD
14814                                           Kanpur
14815  Tirunelveli Eral Tirchchndr Thisayanvilai Peik…
14816                                   Hospet Sandur

                                       destination_city  \
0                                        Kanpur Gurgaon
1                                  Doddablpur Chikblapur
3                                               Mumbai
4                                 Hospet Sandur Bellary
5                                             Chennai
…                                                 …
14812                                Zirakpur Chandigarh
14813                                         Faridabad
14814                                           Kanpur
14815  Eral Tirchchndr Thisayanvilai Peikulam Tirunel…
14816                                   Sandur Bellary

                                     source_city_state  \
0            Bhopal Madhya Pradesh Kanpur Uttar Pradesh
1                 Tumkur Karnataka Doddablpur Karnataka
3                               Mumbai Hub Maharashtra
```

```
4      Bellary Karnataka Hospet Karnataka Sandur Karn…
5                                   Chennai Tamil Nadu
…                                                    …
14812            Chandigarh Punjab Chandigarh Chandigarh
14813                                        FBD Haryana
14814                                  Kanpur Uttar Pradesh
14815  Tirunelveli Tamil Nadu Eral Tamil Nadu Tirchch…
14816                   Hospet Karnataka Sandur Karnataka

                                  destination_city_state  \
0                   Kanpur Uttar Pradesh Gurgaon Haryana
1              Doddablpur Karnataka Chikblapur Karnataka
3                                    Mumbai Maharashtra
4      Hospet Karnataka Sandur Karnataka Bellary Karn…
5                                   Chennai Tamil Nadu
…                                                    …
14812                  Zirakpur Punjab Chandigarh Punjab
14813                                Faridabad Haryana
14814                                  Kanpur Uttar Pradesh
14815  Eral Tamil Nadu Tirchchndr Tamil Nadu Thisayan…
14816                 Sandur Karnataka Bellary Karnataka

                       source_state      destination_state  segment_osrm_time  \
0      Madhya Pradesh Uttar Pradesh  Uttar Pradesh Haryana          16.800000
1                         Karnataka              Karnataka           1.083333
3                   Hub Maharashtra            Maharashtra           0.266667
4                         Karnataka              Karnataka           1.916667
5                         Tamil Nadu             Tamil Nadu          0.383333
…                                 …                      …                 …
14812             Punjab Chandigarh                 Punjab           1.033333
14813                       Haryana                Haryana           0.183333
14814                  Uttar Pradesh          Uttar Pradesh           1.466667
14815                     Tamil Nadu             Tamil Nadu           3.683333
14816                     Karnataka              Karnataka           1.116667

       osrm_time  segment_actual_time  actual_time  \
0      12.383333            25.800000    26.033333
1       1.133333             2.350000     2.383333
3       0.250000             0.983333     0.983333
4       1.950000             5.666667     5.683333
5       0.383333             1.000000     1.016667
…          …                  …           …
14812   1.033333             1.366667     1.383333
14813   0.200000             0.350000     0.350000
14814   0.900000             4.683333     4.700000
14815   3.066667             4.300000     4.400000
14816   1.133333             4.566667     4.583333
```

```
       time_taken_btwn_odstart_and_od_end  start_scan_to_end_scan  \
0                                37.668497               37.650000
1                                 3.026865                3.000000
3                                 1.674916                1.666667
4                                11.972484               11.950000
5                                 3.174797                3.150000
…                                    …                       …
14812                             4.300482                4.283333
14813                             1.009842                1.000000
14814                             7.035331                7.016667
14815                             5.808548                5.783333
14816                             5.906793                5.883333

       segment_osrm_distance  actual_distance_to_destination  osrm_distance  \
0                  1320.4733                       824.732854       991.3523
1                    84.1894                        73.186911        85.1110
3                    19.8766                        17.175274        19.6800
4                   146.7919                       127.448500       146.7918
5                    28.0647                        24.597048        28.0647
…                       …                              …              …
14812                64.8551                        57.762332        73.4630
14813                16.0883                        15.513784        16.0882
14814               104.8866                        38.684839        63.2841
14815               223.5324                       134.723836       177.6635
14816                80.5787                        66.081533        80.5787

                                    route_schedule_uuid
0      thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…
1      thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…
3      thanos::sroute:f0176492-a679-4597-8332-bbd1c7f…
4      thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134…
5      thanos::sroute:9bf03170-d0a2-4a3f-aa4d-9aaab3d…
…                                                    …
14812  thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14…
14813  thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769…
14814  thanos::sroute:5609c268-e436-4e0a-8180-3db4a74…
14815  thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a…
14816  thanos::sroute:412fea14-6d1f-4222-8a5f-a517042…

[14160 rows x 18 columns]
```

10 Processing Data for One hot encoding

[114]:
```
trip_records_without_outliers["destination_source_locations"] =␣
 ↪trip_records_without_outliers["source_city_state"]+"␣
 ↪"+trip_records_without_outliers["destination_city_state"]
trip_records_without_outliers.
 ↪drop(["source_city_state","destination_city_state"],axis = 1,inplace=True)
```

[115]:
```
sc_dc = trip_records_without_outliers.
 ↪groupby(["destination_source_locations"])["trip_uuid"].nunique().
 ↪sort_values(ascending= False).reset_index()
```

[117]:
```
def get_cat(H):
    if 0 <= H <= 50:
        return "Category 7"
    elif 51 <= H <= 100:
        return "Category 6"
    elif 101 <= H <= 200:
        return "Category 5"
    elif 201 <= H <= 300:
        return "Category 4"
    elif 301 <= H <= 400:
        return "Category 3"
    elif 401 <= H <= 500:
        return "Category 2"
    else:
        return "Category 1"
```

[118]:
```
sc_dc["city"]  = pd.Series(map(get_cat,sc_dc["trip_uuid"]))
trip_records_for_encoding = sc_dc.merge(trip_records_without_outliers,
            on="destination_source_locations")
trip_records_for_encoding.
 ↪drop(["destination_source_locations","trip_uuid_x"],axis = 1,inplace=True)
trip_records_for_encoding.drop(["trip_uuid_y"],axis = 1,inplace=True)
# trip_records_for_encoding.sample(15)
encoded_data = pd.get_dummies(trip_records_for_encoding,
             columns=["route_type","city"] )
encoded_data
```

[118]:
```
                                      source_city  \
0                                       Bengaluru
1                                       Bengaluru
2                                       Bengaluru
3                                       Bengaluru
4                                       Bengaluru
…                                             …
14155  Hyderabad Kadthal Kalwakurthy Devarakonda
```

```
14156                          Hyderabad Kadthal
14157                  Hyderabad Kadthal Haliya
14158                  Hyderabad Kadthal Haliya
14159                                       nan

                                destination_city source_state destination_state  \
0                                      Bengaluru    Karnataka         Karnataka
1                                      Bengaluru    Karnataka         Karnataka
2                                      Bengaluru    Karnataka         Karnataka
3                                      Bengaluru    Karnataka         Karnataka
4                                      Bengaluru    Karnataka         Karnataka
…                                                …           …                 …
14155  Kadthal Kalwakurthy Devarakonda Haliya    Telangana         Telangana
14156                    Kadthal Devarakonda    Telangana         Telangana
14157          Kadthal Kalwakurthy Hyderabad    Telangana         Telangana
14158          Kadthal Devarakonda Hyderabad    Telangana         Telangana
14159                                      nan          nan               nan

       segment_osrm_time  osrm_time  segment_actual_time  actual_time  \
0               1.383333   0.950000             3.183333     3.233333
1               1.150000   0.883333             2.666667     2.700000
2               1.183333   0.966667             3.316667     3.333333
3               0.700000   0.733333             1.316667     1.316667
4               0.783333   0.666667             1.750000     1.766667
…                  …        …                    …           …
14155           1.966667   1.983333             3.233333     3.250000
14156           1.483333   1.433333             2.716667     2.750000
14157           2.916667   2.866667             4.950000     4.983333
14158           3.383333   3.333333            10.950000    10.966667
14159           0.800000   0.816667             2.116667     2.133333

       time_taken_btwn_odstart_and_od_end  start_scan_to_end_scan  …  \
0                                4.407028                4.400000  …
1                                4.063014                4.050000  …
2                                4.076829                4.066667  …
3                                4.915934                4.900000  …
4                                3.248617                3.233333  …
…                                   …                     …  …
14155                            6.215731                6.183333  …
14156                            5.625200                5.600000  …
14157                            7.741082                7.716667  …
14158                           13.940494               13.916667  …
14159                            2.146146                2.133333  …

                                    route_schedule_uuid  route_type_Carting  \
0      thanos::sroute:09b4c49e-0e89-40b6-99b3-e671400…                   1
1      thanos::sroute:500aa87c-3d54-4159-a296-0b93c15…                   1
```

```
2       thanos::sroute:16a02d06-e6b6-443b-bd98-0a9e4f4…                    1
3       thanos::sroute:5ffb9921-f943-446e-8796-0b06aa2…                    1
4       thanos::sroute:39928fa7-4ce7-4b78-8e00-c56e31d…                    1
…                                                     …                  …
14155   thanos::sroute:36a3bd62-25dc-44c9-957b-8bcf117…                    0
14156   thanos::sroute:36a3bd62-25dc-44c9-957b-8bcf117…                    0
14157   thanos::sroute:36a3bd62-25dc-44c9-957b-8bcf117…                    0
14158   thanos::sroute:36a3bd62-25dc-44c9-957b-8bcf117…                    0
14159   thanos::sroute:cfb575b8-df26-48f5-8427-6f48f9d…                    0

       route_type_FTL  city_Category 1  city_Category 2  city_Category 3  \
0                   0                1                0                0
1                   0                1                0                0
2                   0                1                0                0
3                   0                1                0                0
4                   0                1                0                0
…                 …                …                …                …
14155               1                0                0                0
14156               1                0                0                0
14157               1                0                0                0
14158               1                0                0                0
14159               1                0                0                0

       city_Category 4  city_Category 5  city_Category 6  city_Category 7
0                    0                0                0                0
1                    0                0                0                0
2                    0                0                0                0
3                    0                0                0                0
4                    0                0                0                0
…                  …                …                …                …
14155                0                0                0                1
14156                0                0                0                1
14157                0                0                0                1
14158                0                0                0                1
14159                0                0                0                1

[14160 rows x 23 columns]
```

10.1 Column Standardization

[119]:
```
['segment_osrm_time', 'osrm_time',
       'segment_actual_time', 'actual_time',
       'time_taken_btwn_odstart_and_od_end', 'start_scan_to_end_scan'␣
 ↪,'segment_osrm_distance', 'actual_distance_to_destination','osrm_distance' ]
```

[119]:
```
['segment_osrm_time',
 'osrm_time',
```

```
 'segment_actual_time',
 'actual_time',
 'time_taken_btwn_odstart_and_od_end',
 'start_scan_to_end_scan',
 'segment_osrm_distance',
 'actual_distance_to_destination',
 'osrm_distance']
```

[120]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
```

[121]:
```
scaler = StandardScaler()
std_data = scaler.fit_transform(encoded_data[['segment_osrm_time',
 'osrm_time',
 'segment_actual_time',
 'actual_time',
 'time_taken_btwn_odstart_and_od_end',
 'start_scan_to_end_scan',
 'segment_osrm_distance',
 'actual_distance_to_destination',
 'osrm_distance']])
std_data = pd.DataFrame(std_data, columns=['segment_osrm_time',
 'osrm_time',
 'segment_actual_time',
 'actual_time',
 'time_taken_btwn_odstart_and_od_end',
 'start_scan_to_end_scan',
 'segment_osrm_distance',
 'actual_distance_to_destination',
 'osrm_distance'])
std_data.head()
```

[121]:

	segment_osrm_time	osrm_time	segment_actual_time	actual_time	\
0	-0.269133	-0.409683	-0.220225	-0.214843	
1	-0.359785	-0.438916	-0.324535	-0.321822	
2	-0.346835	-0.402374	-0.193306	-0.194785	
3	-0.534615	-0.504692	-0.597087	-0.599297	
4	-0.502239	-0.533926	-0.509601	-0.509034	

	time_taken_btwn_odstart_and_od_end	start_scan_to_end_scan	\
0	-0.394178	-0.391956	
1	-0.445632	-0.444397	
2	-0.443566	-0.441900	
3	-0.318061	-0.317039	
4	-0.567441	-0.566761	

	segment_osrm_distance	actual_distance_to_destination	osrm_distance

```
0              -0.362747                       -0.450888      -0.468190
1              -0.448864                       -0.542288      -0.521446
2              -0.416136                       -0.451494      -0.414618
3              -0.536543                       -0.516196      -0.529763
4              -0.549293                       -0.536356      -0.565995
```

[122]:
```
scaler = MinMaxScaler()
MinMax_data = scaler.
 ↪fit_transform(encoded_data[['segment_osrm_time','osrm_time','segment_actual_time','actual_t
␣
 ↪'time_taken_btwn_odstart_and_od_end','start_scan_to_end_scan','segment_osrm_distance','actu
 'osrm_distance']])
MinMax_data = pd.DataFrame(MinMax_data,columns=['segment_osrm_time',
␣
 ↪'osrm_time','segment_actual_time','actual_time','time_taken_btwn_odstart_and_od_end','start
 'segment_osrm_distance','actual_distance_to_destination','osrm_distance'])
MinMax_data.head()
```

[122]:
```
   segment_osrm_time  osrm_time  segment_actual_time  actual_time  \
0           0.069369   0.059302             0.098113     0.098719
1           0.056757   0.054651             0.081402     0.081644
2           0.058559   0.060465             0.102426     0.101921
3           0.032432   0.044186             0.037736     0.037353
4           0.036937   0.039535             0.051752     0.051761

   time_taken_btwn_odstart_and_od_end  start_scan_to_end_scan  \
0                            0.098792                0.098811
1                            0.090329                0.090201
2                            0.090669                0.090611
3                            0.111311                0.111111
4                            0.070296                0.070111

   segment_osrm_distance  actual_distance_to_destination  osrm_distance
0               0.046420                        0.031804       0.036747
1               0.034665                        0.018854       0.028743
2               0.039132                        0.031718       0.044799
3               0.022697                        0.022551       0.027493
4               0.020957                        0.019694       0.022047
```

[123]:
```
std_data
```

[123]:
```
       segment_osrm_time  osrm_time  segment_actual_time  actual_time  \
0              -0.269133  -0.409683            -0.220225    -0.214843
1              -0.359785  -0.438916            -0.324535    -0.321822
2              -0.346835  -0.402374            -0.193306    -0.194785
3              -0.534615  -0.504692            -0.597087    -0.599297
4              -0.502239  -0.533926            -0.509601    -0.509034
```

```
…                …        …                …          …
14155          -0.042502   0.043440          -0.210131    -0.211500
14156          -0.230282  -0.197738          -0.314441    -0.311792
14157           0.326583   0.430787           0.136448     0.136179
14158           0.507888   0.635424           1.347789     1.336342
14159          -0.495764  -0.468150          -0.435575    -0.435486

       time_taken_btwn_odstart_and_od_end  start_scan_to_end_scan  \
0                               -0.394178               -0.391956
1                               -0.445632               -0.444397
2                               -0.443566               -0.441900
3                               -0.318061               -0.317039
4                               -0.567441               -0.566761
…                                     …                       …
14155                           -0.123651               -0.124754
14156                           -0.211977               -0.212156
14157                            0.104495                0.104990
14158                            1.031740                1.033953
14159                           -0.732338               -0.731577

       segment_osrm_distance  actual_distance_to_destination  osrm_distance
0                  -0.362747                       -0.450888      -0.468190
1                  -0.448864                       -0.542288      -0.521446
2                  -0.416136                       -0.451494      -0.414618
3                  -0.536543                       -0.516196      -0.529763
4                  -0.549293                       -0.536356      -0.565995
…                        …                               …              …
14155               0.107675                        0.278861       0.169418
14156              -0.123317                       -0.165131      -0.097018
14157               0.360386                        0.434538       0.512552
14158               0.662356                        0.578210       0.760187
14159              -0.439871                       -0.392780      -0.425349

[14160 rows x 9 columns]
```

[124]:
```
one_hot_encoded_data =␣
 ↪encoded_data[["route_type_Carting","route_type_FTL","city_Category 1",
 "city_Category 2","city_Category 3","city_Category 4",
 "city_Category 5","city_Category 6","city_Category 7"]]
```

[125]:
```
Standardized_Data = pd.concat([std_data,one_hot_encoded_data],axis = 1)
```

[126]:
```
Min_Max_Scaled_Data = pd.concat([MinMax_data,one_hot_encoded_data],axis = 1)
```

[127]:
```
Standardized_Data.sample(5)
```

[127]:

	segment_osrm_time	osrm_time	segment_actual_time	actual_time	\
10234	-0.320934	-0.292748	-0.469223	-0.468917	
3974	-0.651168	-0.650861	-0.704762	-0.702932	
9919	0.870498	1.000845	1.418451	1.416576	
8316	-0.651168	-0.643553	-0.694667	-0.692903	
3001	-0.631742	-0.628936	-0.718221	-0.716304	

	time_taken_btwn_odstart_and_od_end	start_scan_to_end_scan	\
10234	-0.625232	-0.626694	
3974	-0.505355	-0.504330	
9919	0.720208	0.721801	
8316	-0.871661	-0.871420	
3001	-0.266610	-0.264598	

	segment_osrm_distance	actual_distance_to_destination	osrm_distance	\
10234	-0.355372	-0.329361	-0.335589	
3974	-0.637002	-0.616620	-0.640759	
9919	0.803225	0.905757	0.949365	
8316	-0.605569	-0.578325	-0.606410	
3001	-0.591901	-0.585483	-0.591475	

	route_type_Carting	route_type_FTL	city_Category 1	city_Category 2	\
10234	1	0	0	0	
3974	1	0	0	0	
9919	0	1	0	0	
8316	1	0	0	0	
3001	1	0	0	0	

	city_Category 3	city_Category 4	city_Category 5	city_Category 6	\
10234	0	0	0	0	
3974	0	0	1	0	
9919	0	0	0	0	
8316	0	0	0	0	
3001	0	1	0	0	

	city_Category 7
10234	1
3974	0
9919	1
8316	1
3001	0

[128]:
```
Min_Max_Scaled_Data.sample(5)
```

[128]:

	segment_osrm_time	osrm_time	segment_actual_time	actual_time	\
11395	0.223423	0.277907	0.245283	0.243330	
2711	0.009009	0.012791	0.033423	0.033084	

```
5237            0.016216   0.022093             0.084097     0.083244
7703            0.041441   0.033721             0.074394     0.073639
1831            0.003604   0.005814             0.014555     0.014408

      time_taken_btwn_odstart_and_od_end  start_scan_to_end_scan  \
11395                            0.247433                0.246822
2711                             0.049367                0.049200
5237                             0.150690                0.150882
7703                             0.060118                0.060271
1831                             0.027142                0.027060

      segment_osrm_distance  actual_distance_to_destination  osrm_distance  \
11395              0.194325                        0.207660       0.220893
2711               0.008449                        0.007688       0.010166
5237               0.014512                        0.012598       0.016727
7703               0.033375                        0.028983       0.031391
1831               0.003288                        0.001643       0.003956

      route_type_Carting  route_type_FTL  city_Category 1  city_Category 2  \
11395                  0               1                0                0
2711                   1               0                0                0
5237                   1               0                0                0
7703                   1               0                0                0
1831                   1               0                0                0

      city_Category 3  city_Category 4  city_Category 5  city_Category 6  \
11395                0                0                0                0
2711                 0                1                0                0
5237                 0                0                0                1
7703                 0                0                0                0
1831                 1                0                0                0

      city_Category 7
11395                1
2711                 0
5237                 0
7703                 1
1831                 0
```

11 Route analysis

[129]:
```
A = data.groupby("route_schedule_uuid")["route_type"].unique().reset_index()
B = data.groupby("route_schedule_uuid")["destination_city"].unique().
  ↪reset_index()
B.columns = ["route_schedule_uuid","destination_cities"]
C = data.groupby("route_schedule_uuid")["source_city"].unique().reset_index()
```

```
C.columns = ["route_schedule_uuid","source_cities"]
D = data.groupby("route_schedule_uuid")["source_state"].unique().reset_index()
D.columns = ["route_schedule_uuid","source_states"]
E = data.groupby("route_schedule_uuid")["destination_state"].unique().
 ↪reset_index()
E.columns = ["route_schedule_uuid","destination_states"]
F = data.groupby("route_schedule_uuid")[["source_state",
                                         "destination_state"]].nunique().
 ↪sort_values(by="source_state",
                                                                        ␣
 ↪     ascending=False).reset_index()
F.columns = ["route_schedule_uuid","#source_states"
             ,"#destination_states"]
G = trip_records.
 ↪groupby("route_schedule_uuid")["actual_distance_to_destination"].mean().
 ↪reset_index()
G.columns = ["route_schedule_uuid","Average_Actual_distance_to_destination"]
H = trip_records["route_schedule_uuid"].value_counts().reset_index()
H.columns = ["route_schedule_uuid","Number_of_Trips"]
I = data.groupby("route_schedule_uuid")[["source_city",
                                         "destination_city"]].nunique().
 ↪sort_values(by="source_city",
                                                                        ␣
 ↪     ascending=False).reset_index()
I.columns = ["route_schedule_uuid","#source_cities"
             ,"#destination_cities"]
```

[130]:
```
route_records = I.merge(H.merge(G.merge(F.merge(E.merge(D.merge(C.merge(A.
 ↪merge(B,
       on ="route_schedule_uuid",
       how = "outer"),on ="route_schedule_uuid",
       how = "outer"),
      on ="route_schedule_uuid",
       how = "outer"),
      on ="route_schedule_uuid",
       how = "outer"),
      on ="route_schedule_uuid",
       how = "outer"),
      on ="route_schedule_uuid",
       how = "outer"),
      on ="route_schedule_uuid",
       how = "outer"),on ="route_schedule_uuid",
       how = "outer")
```

[131]:
```
route_records.isna().sum()
```

[131]:
```
route_schedule_uuid                          0
#source_cities                               0
#destination_cities                          0
Number_of_Trips                              0
Average_Actual_distance_to_destination       0
#source_states                               0
#destination_states                          0
destination_states                           0
source_states                                0
source_cities                                0
route_type                                   0
destination_cities                           0
dtype: int64
```

[132]:
```
route_records.dropna(inplace=True)
```

[133]:
```
route_records["route_type"] = route_records["route_type"].astype("str").str.
 ↪strip("[]").str.replace("'","")
route_records["source_cities"] = route_records["source_cities"].astype("str").
 ↪str.strip("[]").str.replace("'","")
route_records["destination_cities"] = route_records["destination_cities"].
 ↪astype("str").str.strip("[]").str.replace("'","")
route_records["source_states"] = route_records["source_states"].astype("str").
 ↪str.strip("[]").str.replace("'","")

route_records["destination_states"] = route_records["destination_states"].
 ↪astype("str").str.strip("[]").str.replace("'","")
```

[134]:
```
route_records
```

[134]:

	route_schedule_uuid	#source_cities	\
0	thanos::sroute:d010efca-d90d-4977-b987-eae68c5…	13	
1	thanos::sroute:4cbecb35-356b-4b68-bf3c-6225b5e…	10	
2	thanos::sroute:ae5c430f-6153-48d1-8fe5-d5f0bbc…	10	
3	thanos::sroute:f8968c72-5222-4d81-9eed-8a6d88f…	9	
4	thanos::sroute:ed5b80be-7abf-424d-b8cd-d81556a…	9	
…	…	…	
1499	thanos::sroute:9e7bb811-593f-47bc-ac49-ba03ed8…	1	
1500	thanos::sroute:46b9641b-55b5-4b15-b039-2612a50…	1	
1501	thanos::sroute:b48f633d-15cb-4744-a0b9-21df0a9…	1	
1502	thanos::sroute:265efe06-3625-4fba-afee-07b5b64…	0	
1503	thanos::sroute:cfb575b8-df26-48f5-8427-6f48f9d…	0	

	#destination_cities	Number_of_Trips	\
0	11	14	
1	10	12	
2	10	20	

```
3                       9                9
4                       8               20
…                     …                …
1499                    1               19
1500                    1               15
1501                    1                7
1502                    1                1
1503                    0                1

      Average_Actual_distance_to_destination  #source_states  \
0                                 281.596486              2
1                                 332.602225              2
2                                 351.611796              1
3                                 195.257193              1
4                                 178.737233              1
…                                        …              …
1499                               17.617532              1
1500                               10.137219              1
1501                               15.467701              1
1502                              236.815038              0
1503                               50.844665              0

      #destination_states       destination_states            source_states  \
0                       2  Assam Arunachal Pradesh  Assam Arunachal Pradesh
1                       2          Assam Meghalaya          Assam Meghalaya
2                       1                Rajasthan                Rajasthan
3                       2            Karnataka Goa                Karnataka
4                       1                Rajasthan                Rajasthan
…                     …                        …                        …
1499                    1              Maharashtra              Maharashtra
1500                    1              Maharashtra              Maharashtra
1501                    1                Karnataka                Karnataka
1502                    1             Uttar Pradesh                      nan
1503                    0                      nan                      nan

                                        source_cities route_type  \
0     Guwahati LakhimpurN Dhemaji Likabali Tezpur Pa…        FTL
1     Guwahati Rangia Kokrajhar Dhubri Bilasipara Tu…        FTL
2     Jaipur Chomu Reengus Sikar Bikaner Didwana Suj…        FTL
3     Mangalore Udupi Kundapura Bhatkal Honnavar Kum…        FTL
4     Ajmer Beawar Bilara Bijainagar Kekri Nasirabad…        FTL
…                                                   …        …
1499                                           Mumbai    Carting
1500                                           Mumbai    Carting
1501                                        Bengaluru    Carting
1502                                              nan        FTL
1503                                              nan        FTL
```

```
                                        destination_cities
0     Tezpur Dhemaji Silapathar Pasighat Mangaldoi I…
1     Rangia Nalbari Dhubri Bilasipara Lakhipur Guwa…
2     Chomu Reengus Sikar Bikaner Nokha Sujangarh Ja…
3     Uchila Kundapura Bhatkal Honnavar Kumta Ankola…
4     Beawar Bilara Badnaur Kekri Nasirabad Ajmer Bi…
…                                                   …
1499                                           Mumbai
1500                                           Mumbai
1501                                        Bengaluru
1502                                         Mainpuri
1503                                              nan

[1504 rows x 12 columns]
```

[135]:
```
route_records["ROUTE"] = route_records["source_cities"] + " -- " +␣
 ↪route_records["destination_cities"]
route_records.drop(["route_schedule_uuid"],axis = 1,inplace=True)
first_column = route_records.pop('ROUTE')
route_records.insert(0, 'ROUTE', first_column)
route_records["SouceToDestination_city"] = route_records["source_cities"].str.
 ↪split(" ").apply(lambda x:x[0]) +" TO " +route_records["destination_cities"].
 ↪str.split(" ").apply(lambda x:x[-1])
first_column = route_records.pop('SouceToDestination_city')
route_records.insert(0, 'SouceToDestination_city', first_column)
route_records
```

[135]:
```
     SouceToDestination_city  \
0     Guwahati TO LakhimpurN
1           Guwahati TO Tura
2           Jaipur TO Tarnau
3         Mangalore TO Udupi
4            Ajmer TO Raipur
…                        …
1499        Mumbai TO Mumbai
1500        Mumbai TO Mumbai
1501  Bengaluru TO Bengaluru
1502         nan TO Mainpuri
1503              nan TO nan

                                                  ROUTE  #source_cities  \
0     Guwahati LakhimpurN Dhemaji Likabali Tezpur Pa…               13
1     Guwahati Rangia Kokrajhar Dhubri Bilasipara Tu…               10
2     Jaipur Chomu Reengus Sikar Bikaner Didwana Suj…               10
3     Mangalore Udupi Kundapura Bhatkal Honnavar Kum…                9
4     Ajmer Beawar Bilara Bijainagar Kekri Nasirabad…                9
```

```
…                                          …             …
1499                              Mumbai -- Mumbai             1
1500                              Mumbai -- Mumbai             1
1501                          Bengaluru -- Bengaluru           1
1502                                nan -- Mainpuri            0
1503                                     nan -- nan            0

      #destination_cities  Number_of_Trips  \
0                      11              14
1                      10              12
2                      10              20
3                       9               9
4                       8              20
…                     …               …
1499                    1              19
1500                    1              15
1501                    1               7
1502                    1               1
1503                    0               1

      Average_Actual_distance_to_destination  #source_states  \
0                                 281.596486              2
1                                 332.602225              2
2                                 351.611796              1
3                                 195.257193              1
4                                 178.737233              1
…                                      …             …
1499                               17.617532              1
1500                               10.137219              1
1501                               15.467701              1
1502                              236.815038              0
1503                               50.844665              0

      #destination_states       destination_states            source_states  \
0                       2  Assam Arunachal Pradesh  Assam Arunachal Pradesh
1                       2          Assam Meghalaya          Assam Meghalaya
2                       1                Rajasthan                Rajasthan
3                       2            Karnataka Goa                Karnataka
4                       1                Rajasthan                Rajasthan
…                     …                        …                        …
1499                    1              Maharashtra              Maharashtra
1500                    1              Maharashtra              Maharashtra
1501                    1                Karnataka                Karnataka
1502                    1             Uttar Pradesh                      nan
1503                    0                      nan                      nan

                                    source_cities route_type  \
```

```
0     Guwahati LakhimpurN Dhemaji Likabali Tezpur Pa…        FTL
1     Guwahati Rangia Kokrajhar Dhubri Bilasipara Tu…        FTL
2     Jaipur Chomu Reengus Sikar Bikaner Didwana Suj…        FTL
3     Mangalore Udupi Kundapura Bhatkal Honnavar Kum…        FTL
4     Ajmer Beawar Bilara Bijainagar Kekri Nasirabad…        FTL
…                                                  …        …
1499                                          Mumbai    Carting
1500                                          Mumbai    Carting
1501                                       Bengaluru    Carting
1502                                             nan        FTL
1503                                             nan        FTL

                                    destination_cities
0     Tezpur Dhemaji Silapathar Pasighat Mangaldoi I…
1     Rangia Nalbari Dhubri Bilasipara Lakhipur Guwa…
2     Chomu Reengus Sikar Bikaner Nokha Sujangarh Ja…
3     Uchila Kundapura Bhatkal Honnavar Kumta Ankola…
4     Beawar Bilara Badnaur Kekri Nasirabad Ajmer Bi…
…                                                  …
1499                                          Mumbai
1500                                          Mumbai
1501                                       Bengaluru
1502                                        Mainpuri
1503                                             nan

[1504 rows x 13 columns]
```

12 Exploratory Data Analysis : (getting some insights from pre-processed data)

Busiest Route Analysis :

Number of Trips between cities , sorted highest to lowest

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

[136]:
```
Number_of_trips_between_cities = data.groupby(["source_city_state",
                                               ␣
 ↪"destination_city_state"])["trip_uuid"].nunique().
 ↪sort_values(ascending=False).reset_index()
Number_of_trips_between_cities.head(25)
```

[136]:
```
          source_city_state destination_city_state  trip_uuid
0      Bengaluru Karnataka    Bengaluru Karnataka       1369
1     Bhiwandi Maharashtra     Mumbai Maharashtra        512
2       Mumbai Maharashtra     Mumbai Maharashtra        361
3      Hyderabad Telangana    Hyderabad Telangana        308
```

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

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

[137]:
```
Number_of_trips_between_cities.
↪loc[Number_of_trips_between_cities["source_city_state"] !=␣
↪Number_of_trips_between_cities["destination_city_state"]].head(25)
```

[137]:

	source_city_state	destination_city_state	trip_uuid
1	Bhiwandi Maharashtra	Mumbai Maharashtra	512
4	Mumbai Maharashtra	Bhiwandi Maharashtra	282
5	Delhi Delhi	Gurgaon Haryana	248
6	Gurgaon Haryana	Delhi Delhi	237
7	Mumbai Hub Maharashtra	Mumbai Maharashtra	227
9	MAA Tamil Nadu	Chennai Tamil Nadu	204
10	Chennai Tamil Nadu	MAA Tamil Nadu	141
11	Bengaluru Karnataka	HBR Karnataka	133
13	Pune Maharashtra	PNQ Maharashtra	122
16	Pune Maharashtra	Bhiwandi Maharashtra	107
18	Chandigarh Chandigarh	Chandigarh Punjab	100
19	Kolkata West Bengal	CCU West Bengal	96
20	Gurgaon Haryana	Sonipat Haryana	92
21	Sonipat Haryana	Gurgaon Haryana	86
22	Chandigarh Punjab	Chandigarh Chandigarh	84
23	HBR Karnataka	Bengaluru Karnataka	79

```
24     Bengaluru Karnataka          BLR Karnataka         78
26               Del Delhi        Gurgaon Haryana         76
27    Bhiwandi Maharashtra       Pune Maharashtra         72
28         Ludhiana Punjab     Chandigarh Punjab         71
30       Chandigarh Punjab        Gurgaon Haryana         66
31         Gurgaon Haryana    Bengaluru Karnataka         66
32  LowerParel Maharashtra     Mumbai Maharashtra         65
34  Mumbai Hub Maharashtra   Bhiwandi Maharashtra         63
35         PNQ Maharashtra       Pune Maharashtra         62
```

If we talk about , not having equal source and destination states , source and destination cities having higest number of trips in between are :

delhi to gurgao Gurgaon,Haryana TO Bengaluru,Karnataka Bhiwandi/Mumbai,Maharashtra TO Pune Maharashtra Sonipat TO Gurgaon,Haryana

it is also been observed that lots of deliveries are happening to airpots like : Chennai to MAA chennai international Airport , Pune to Pune Airport (PNQ),Kolkata to CCU West Bengal Kolkata International Airport , Bengluru to BLR-Bengaluru Internation Airport etc.

[139]:
```
route_records[["ROUTE","Number_of_Trips",
               "Average_Actual_distance_to_destination",
               "#source_cities",
               "#destination_cities"]].
↪sort_values(by="Number_of_Trips",ascending=False).head(25)
```

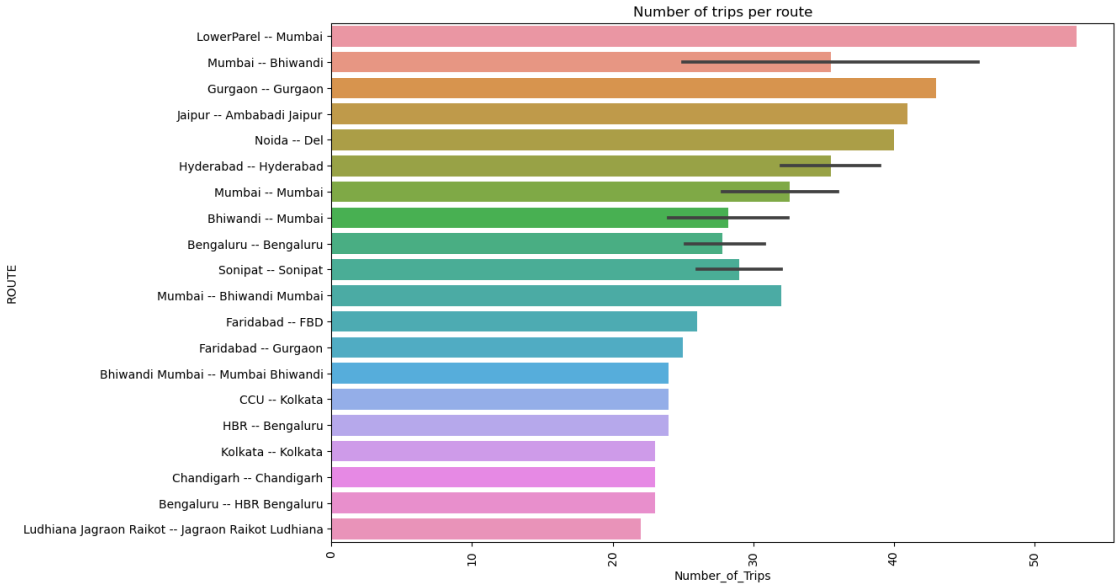
[139]:
```
                          ROUTE  Number_of_Trips  \
1465       LowerParel -- Mumbai               53
1426         Mumbai -- Bhiwandi               46
808          Gurgaon -- Gurgaon               43
679   Jaipur -- Ambabadi Jaipur               41
1257              Noida -- Del               40
1368     Hyderabad -- Hyderabad               39
1273           Mumbai -- Mumbai               37
1359           Mumbai -- Mumbai               36
1303         Bhiwandi -- Mumbai               35
700            Mumbai -- Mumbai               34
751            Mumbai -- Mumbai               33
1060     Bengaluru -- Bengaluru               33
793          Sonipat -- Sonipat               32
972      Hyderabad -- Hyderabad               32
1184  Mumbai -- Bhiwandi Mumbai               32
874      Bengaluru -- Bengaluru               30
1177         Bhiwandi -- Mumbai               30
1354     Bengaluru -- Bengaluru               27
921            Faridabad -- FBD               26
1480         Sonipat -- Sonipat               26
1041         Mumbai -- Bhiwandi               25
877        Faridabad -- Gurgaon               25
```

```
833          Bhiwandi -- Mumbai                25
1249     Bengaluru -- Bengaluru                25
869      Bengaluru -- Bengaluru                24

      Average_Actual_distance_to_destination  #source_cities  \
1465                               16.428868              1
1426                               20.199445              1
808                                29.740842              1
679                                15.348495              1
1257                               10.882902              1
1368                               35.695641              1
1273                               13.882863              1
1359                               17.526251              1
1303                               21.241534              1
700                                15.906614              1
751                                15.668726              1
1060                               28.067004              1
793                                11.691243              1
972                                21.835579              1
1184                               21.601109              1
874                                28.055789              1
1177                               21.396002              1
1354                               27.967087              1
921                                 9.677121              1
1480                               12.182486              1
1041                               19.942191              1
877                                47.091622              1
833                                21.531705              1
1249                               28.019668              1
869                                41.396497              1

      #destination_cities
1465                    1
1426                    1
808                     1
679                     2
1257                    1
1368                    1
1273                    1
1359                    1
1303                    1
700                     1
751                     1
1060                    1
793                     1
972                     1
1184                    2
```

874	1
1177	1
1354	1
921	1
1480	1
1041	1
877	1
833	1
1249	1
869	1

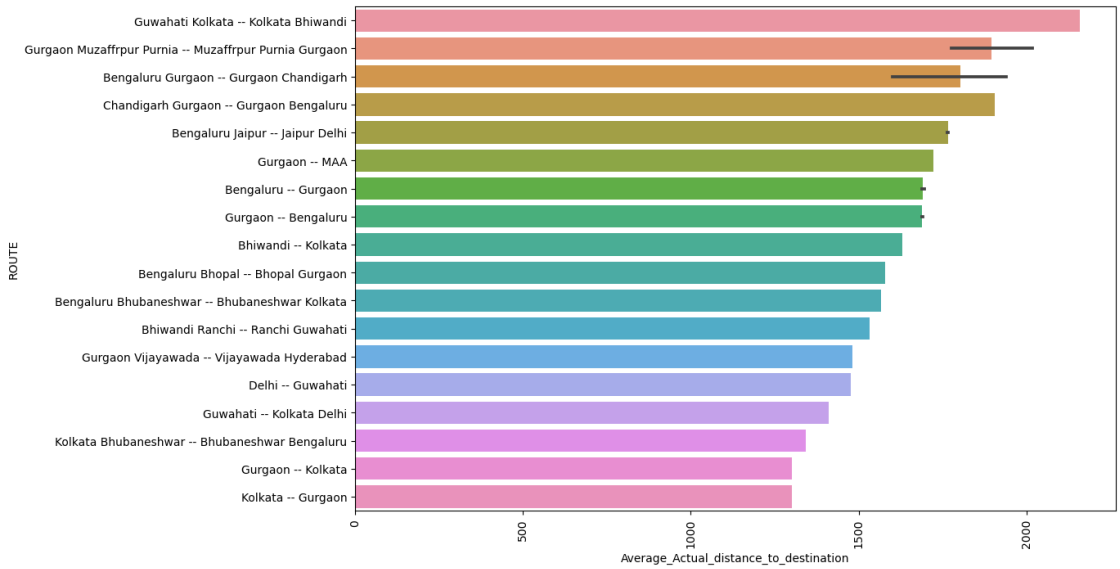
12.1 Top Routes having Maximum Number of Trips between/within the source and destinations

[140]:
```
plt.figure(figsize=(12,8))

X = route_records[["ROUTE", "Number_of_Trips",
               ]].sort_values(by="Number_of_Trips",ascending=False).head(35)
sns.barplot(y = X["ROUTE"],
            x= X["Number_of_Trips"])
plt.title("Number of trips per route")
plt.xticks(rotation = 90)
plt.show()
```



[141]:
```
plt.figure(figsize=(12,8))

X = route_records[["ROUTE", "Average_Actual_distance_to_destination",
```

```
                ]].
 ↪sort_values(by="Average_Actual_distance_to_destination",ascending=False).
 ↪head(25)
sns.barplot(y = X["ROUTE"],
            x = X["Average_Actual_distance_to_destination"])
plt.xticks(rotation = 90)
plt.show()
```



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

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

12.2 Busiest and Longest Routes

[142]:
```
Busiest_and_Longest_Routes  =␣
 ↪route_records[(route_records["Average_Actual_distance_to_destination"] >␣
 ↪route_records["Average_Actual_distance_to_destination"].quantile(0.75))
             & (route_records["Number_of_Trips"] >␣
 ↪route_records["Number_of_Trips"].quantile(0.75))].
 ↪sort_values(by="Average_Actual_distance_to_destination"
                                                                    ␣
 ↪                                    ,ascending=False)
Busiest_and_Longest_Routes_top25 = Busiest_and_Longest_Routes[["source_cities",
                                                                ␣
 ↪"destination_cities",
```

```
 ↪"Number_of_Trips",
                                                                    ␣
 ↪"Average_Actual_distance_to_destination"]].head(25)
Busiest_and_Longest_Routes_top25
```

[142]:

	source_cities	destination_cities	\
629	Chandigarh Gurgaon	Gurgaon Bengaluru	
995	Gurgaon	Bengaluru	
991	Gurgaon	Bengaluru	
512	Bengaluru Bhubaneshwar	Bhubaneshwar Kolkata	
745	Guwahati	Kolkata Delhi	
624	Kolkata Bhubaneshwar	Bhubaneshwar Bengaluru	
752	Gurgaon	Kolkata	
588	Delhi Gurgaon	Gurgaon Kolkata	
826	Gurgaon	Hyderabad	
541	Chandigarh Gurgaon	Gurgaon Bhiwandi	
442	Delhi Gurgaon	Gurgaon Pune	
445	Bhiwandi Sonipat	Sonipat Chandigarh	
739	Pune	Gurgaon	
1377	Bhiwandi	Delhi	
1049	Delhi	Bhiwandi	
313	Bengaluru Kolhapur Surat	Kolhapur Surat Ahmedabad	
1219	Gurgaon	Bhiwandi	
197	Sasaram Kanpur Kolkata Dhanbad	Kanpur Gurgaon Dhanbad Sasaram	
1136	Gurgaon	Ranchi	
1286	Surat	Delhi	
439	Kolkata Ranchi	Ranchi Gurgaon	
1108	Gurgaon	Sasaram	
1454	Gurgaon	Ahmedabad	
223	Bhopal Kanpur Auraiya Etawah	Kanpur Auraiya Etawah Gurgaon	
863	Bhiwandi	Hyderabad	

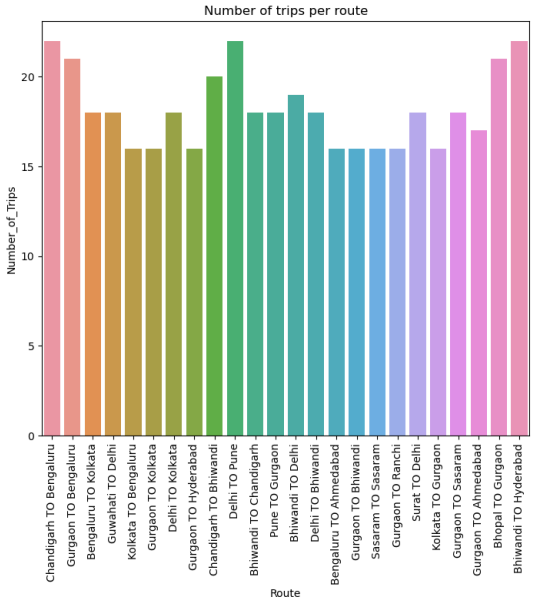
	Number_of_Trips	Average_Actual_distance_to_destination
629	22	1905.766051
995	21	1689.873158
991	21	1689.791894
512	18	1567.577507
745	18	1411.208424
624	16	1342.143081
752	16	1300.572161
588	18	1263.113211
826	16	1236.572072
541	20	1170.817927
442	22	1151.514940
445	18	1129.609705
739	18	1120.729446

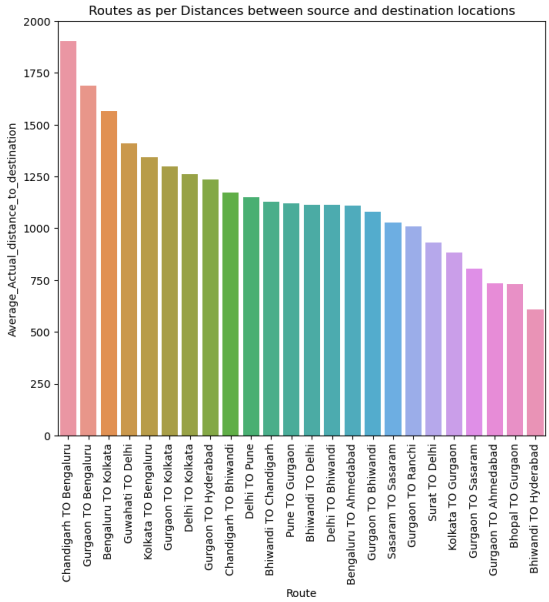
1377	19	1114.214670
1049	18	1114.182197
313	16	1110.015339
1219	16	1078.076312
197	16	1028.024726
1136	16	1010.953223
1286	18	931.980821
439	16	881.621264
1108	18	804.210670
1454	17	735.550450
223	21	731.634456
863	22	607.514619

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

Chandigarh TO Bengaluru Gurgaon TO Bengaluru Bengaluru TO Kolkata Guwahati TO Delhi Delhi TO Kolkata Chandigarh TO Gurgaon Gurgaon TO Hydrabad Benglore TO Ahmedabad Surat TO Delhi Gurgaon TO Ahmedabad

[144]:
```
Busiest_and_Longest_Routes_top25["Route"] =␣
 ↪Busiest_and_Longest_Routes_top25["source_cities"].str.split(" ").
 ↪apply(lambda x:x[0]) + " TO " +␣
 ↪Busiest_and_Longest_Routes_top25["destination_cities"].str.split(" ").
 ↪apply(lambda x:x[-1])
Busiest_and_Longest_Routes_top25.
 ↪drop(["source_cities","destination_cities"],axis = 1,inplace=True)
plt.figure(figsize=(18,7))

plt.subplot(121)
plt.title("Number of trips per route")
sns.barplot(x=Busiest_and_Longest_Routes_top25["Route"],
            y = Busiest_and_Longest_Routes_top25["Number_of_Trips"])
plt.xticks(rotation = 90)
plt.subplot(122)
plt.title("Routes as per Distances between source and destination locations")
sns.barplot(x=Busiest_and_Longest_Routes_top25["Route"],
            y=␣
 ↪Busiest_and_Longest_Routes_top25["Average_Actual_distance_to_destination"])
plt.xticks(rotation = 90)
plt.show()
```





12.3 Routes : passing through maxinum number of cities

[145]:
```
route_records[["SouceToDestination_city","Number_of_Trips",
               "Average_Actual_distance_to_destination",
               "#source_cities",
               "#destination_cities"]].sort_values(by=["#source_cities",
                                                       "#destination_cities",
                                                       "Number_of_Trips"]
                                                   ,ascending=False).head(25)
```

[145]:

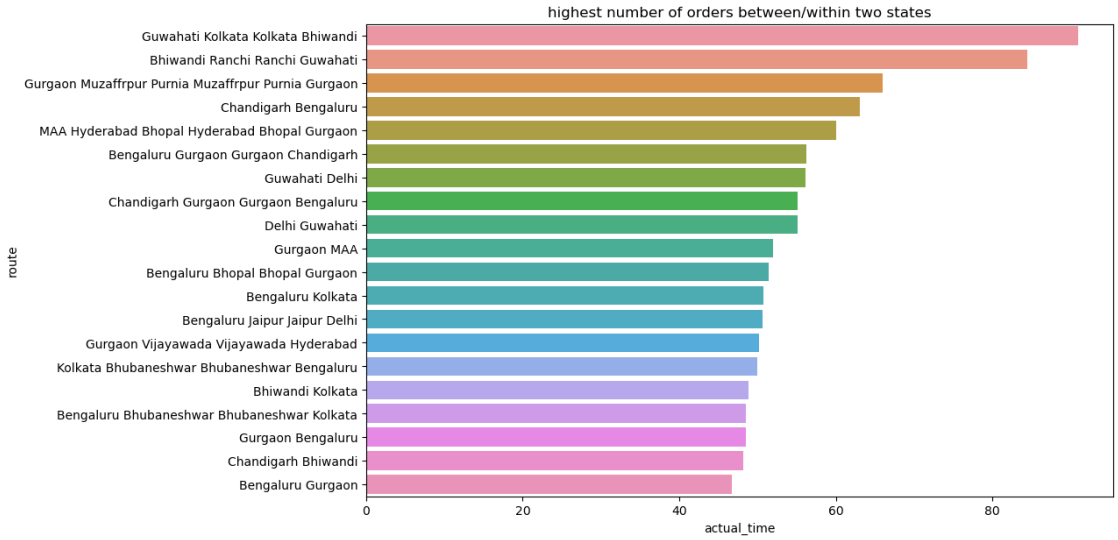
	SouceToDestination_city	Number_of_Trips	\
0	Guwahati TO LakhimpurN	14	
2	Jaipur TO Tarnau	20	
1	Guwahati TO Tura	12	
3	Mangalore TO Udupi	9	
4	Ajmer TO Raipur	20	
5	Mainpuri TO Tilhar	12	
8	Hassan TO Koppa	21	
15	Shrirampur TO Sangamner	20	
7	Musiri TO Tiruchi	19	
9	Bijnor TO Bijnor	17	
10	Dausa TO Lalsot	17	
17	Tinusukia TO Dibrugarh	16	
12	Pondicherry TO Pondicherry	12	
14	Mysore TO Mysore	12	
6	Golaghat TO Guwahati	11	
13	Varanasi TO Varanasi	8	

```
16      Vijayawada TO Suryapet                8
11     Hyderabad TO Miryalguda                7
27       Srikakulam TO Bobbili               22
36         Pukhrayan TO Kanpur               22
48            Dhule TO Shirpur               22
30        Madhupur TO Madhupur               21
38      Kamareddy TO Kamareddy               21
42             Noida TO Khurja               21
20         Junagadh TO Veraval               19

    Average_Actual_distance_to_destination  #source_cities  \
0                                281.596486              13
2                                351.611796              10
1                                332.602225              10
3                                195.257193               9
4                                178.737233               9
5                                207.247057               8
8                                200.497832               7
15                               204.509529               7
7                                219.845121               7
9                                209.400685               7
10                               232.408310               7
17                               111.098543               7
12                               230.253602               7
14                               154.324190               7
6                                258.546587               7
13                                82.545019               7
16                               407.029391               7
11                               420.603709               7
27                               154.495283               6
36                               139.834945               6
48                               150.016233               6
30                               252.072259               6
38                               177.923330               6
42                               208.714043               6
20                               179.538596               6

    #destination_cities
0                    11
2                    10
1                    10
3                     9
4                     8
5                     8
8                     7
15                    7
7                     7
```

9	7
10	7
17	7
12	7
14	7
6	7
13	7
16	7
11	7
27	6
36	6
48	6
30	6
38	6
42	6
20	6

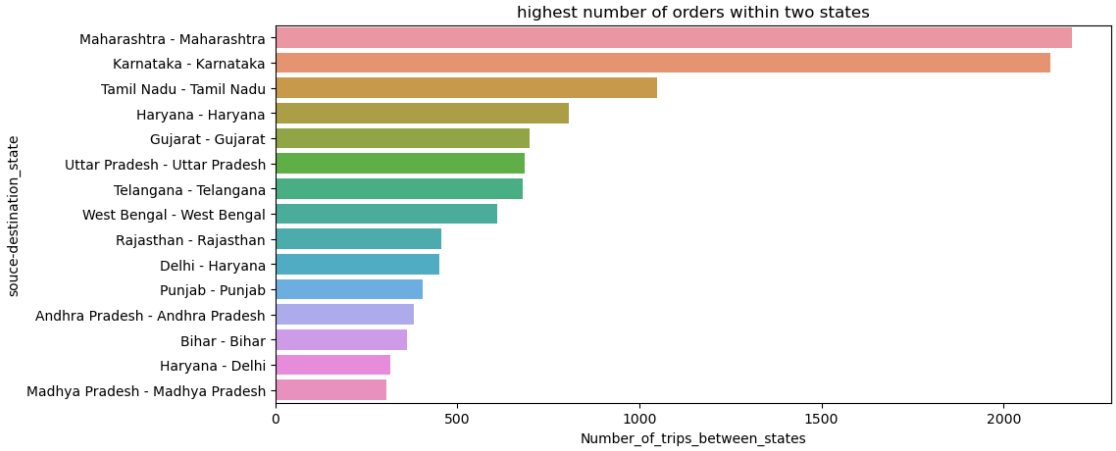
12.4 Top 20 Longest Route as per average actual time taken from one city to another city

[146]:
```
Longest_route_as_per_actual_trip_time = trip_records.groupby(["source_city",
                      "destination_city"])["actual_time"].mean().
 ↪sort_values(ascending=False).head(20).reset_index()
Longest_route_as_per_actual_trip_time["route"] =␣
 ↪Longest_route_as_per_actual_trip_time["source_city"] + " " +␣
 ↪Longest_route_as_per_actual_trip_time["destination_city"]
Longest_route_as_per_actual_trip_time.drop(["source_city",
                                            "destination_city"],axis =␣
 ↪1,inplace=True)
Longest_route_as_per_actual_trip_time
plt.figure(figsize=(11,7))
sns.barplot(y = Longest_route_as_per_actual_trip_time["route"],
           x = Longest_route_as_per_actual_trip_time["actual_time"],)
plt.title("highest number of orders between/within two states")
plt.show()
```

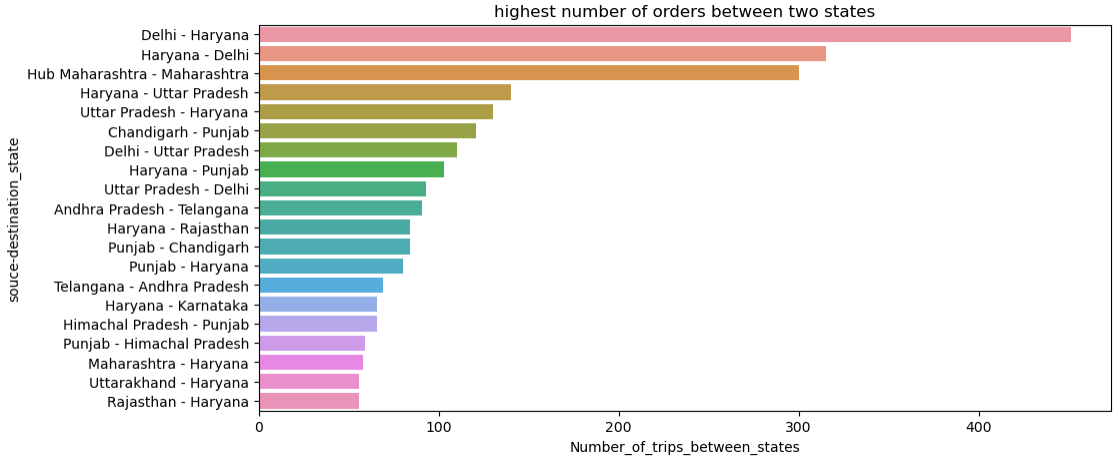


12.5 Highest number of Trips happening between/within two states

[147]:
```
highest_order_between_states = data.groupby(["source_state",
                                             "destination_state"])["trip_uuid"].
 ↪nunique().sort_values(ascending=False).reset_index()
HOBS  = highest_order_between_states.head(15)
HOBS["souce-destination"] = HOBS["source_state"] + " - " +␣
 ↪HOBS["destination_state"]
HOBS.drop(["source_state","destination_state"],axis = 1, inplace=True)
HOBS.columns = ["Number_of_trips_between_states","souce-destination_state"]

plt.figure(figsize=(11,5))
sns.barplot(y = HOBS["souce-destination_state"],
            x = HOBS["Number_of_trips_between_states"],)
plt.title("highest number of orders within two states")
plt.show()
```



[148]:
```
HOBS = data.groupby(["source_state","destination_state"])["trip_uuid"].
 ↪nunique().sort_values(ascending=False).reset_index()
HOBS = HOBS[HOBS["source_state"]!=HOBS["destination_state"]].head(20)

HOBS["souce-destination"] = HOBS["source_state"] + " - " +␣
 ↪HOBS["destination_state"]
HOBS.drop(["source_state","destination_state"],axis = 1, inplace=True)
HOBS.columns = ["Number_of_trips_between_states","souce-destination_state"]

plt.figure(figsize=(11,5))
sns.barplot(y = HOBS["souce-destination_state"],
            x = HOBS["Number_of_trips_between_states"],)
plt.title("highest number of orders between two states")
plt.show()
```

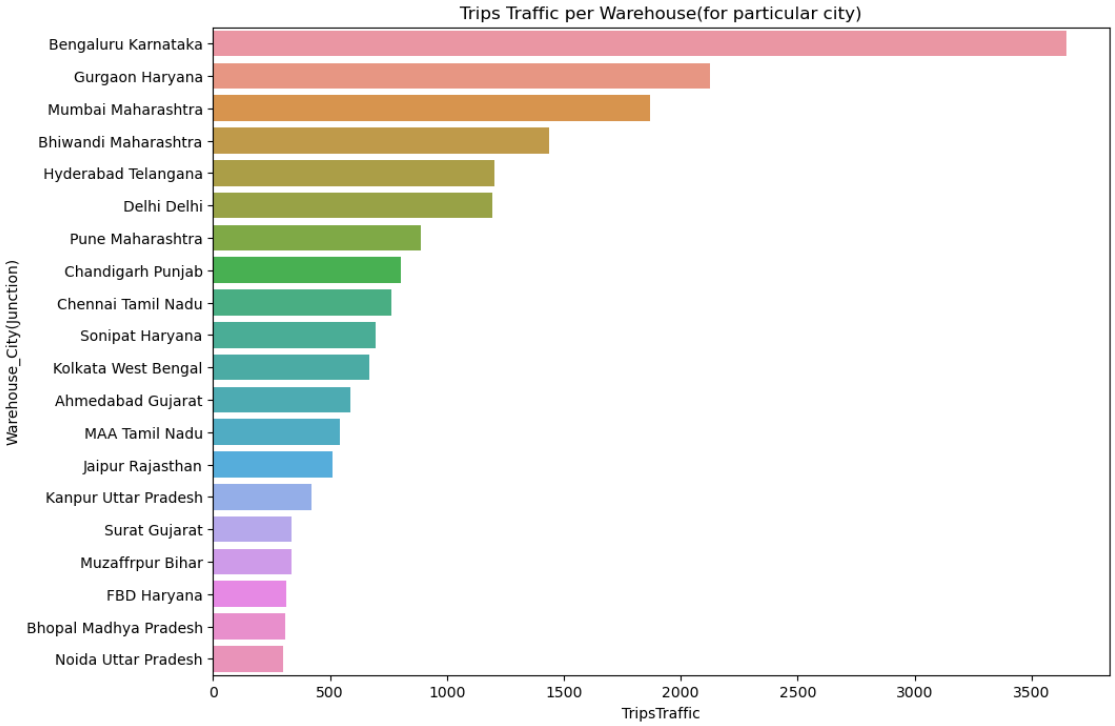


12.6 Top 20 warehouses with heavy traffic

[149]:
```
destination_traffic = data.groupby(["destination_city_state"])["trip_uuid"].
 ↪nunique().reset_index()
source_traffic = data.groupby(["source_city_state"])["trip_uuid"].nunique().
 ↪reset_index()
transactions = source_traffic.merge(destination_traffic,
                                    left_on="source_city_state"
                                    ,right_on="destination_city_state")
transactions.columns =␣
 ↪["source_city_state","#Trips_s","destination_city_state","#Trips_d"]
transactions["TripsTraffic"] = transactions["#Trips_s"]+transactions["#Trips_d"]
transactions.drop(["#Trips_s","#Trips_d","destination_city_state"],axis =␣
 ↪1,inplace=True)
transactions.columns = ["Warehouse_City(Junction)","TripsTraffic"]
```

[150]:
```
T = transactions.sort_values(by=["TripsTraffic"],ascending=False).head(20)
```

[151]:
```
plt.figure(figsize=(11,8))
sns.barplot(y = T["Warehouse_City(Junction)"],
            x = T["TripsTraffic"])
plt.title("Trips Traffic per Warehouse(for particular city)")
plt.show()
```



[152]:
```
trip_records.groupby(["source_state","destination_state"])["trip_uuid"].count().
↪sort_values(ascending=False).head(15).reset_index()
```

[152]:

	source_state	destination_state	trip_uuid
0	Maharashtra	Maharashtra	2085
1	Karnataka	Karnataka	2002
2	Tamil Nadu	Tamil Nadu	996
3	Haryana	Haryana	771
4	Telangana	Telangana	627
5	Gujarat	Gujarat	624
6	West Bengal	West Bengal	610
7	Uttar Pradesh	Uttar Pradesh	529
8	Rajasthan	Rajasthan	400
9	Delhi	Haryana	385
10	Andhra Pradesh	Andhra Pradesh	344
11	Punjab	Punjab	342
12	Bihar	Bihar	330
13	Haryana	Delhi	307
14	Hub Maharashtra	Maharashtra	300

13 Insights

- 14817 different trips happened between source to destinations during 2018 , September and October.
- 1504 delivery routes on which trips are happenig.
- we have 1508 unique source centers and 1481 unique destination centers
- From 14817 total different trips , we have 8908 (60%) of the trip-routes are Carting , which consists of small vehicles and 5909 (40%) of total trip-routes are FTL : which are Full Truck Load get to the destination sooner. as no other pickups or drop offs along the way .

13.0.1 Hypothesis tests Results

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

- Population Mean osrm time is less than Population Mean segment osrm time.
- Average of OSRM distance for population is less than average of segment OSRM distance
- population OSRM estimated distance is higher than the actual distance from source to destination warehouse.

13.0.2 EDA Results

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

14 Recommendations

- As per analysis, It is recommended to use Carting (small vehicles) for delivery with in the city in order to reduce the delivery time, and Heavy trucks for long distance trips or heavy load. based on this , we can optimize the delivery time as well as increase the revenue as per requirements.
- Incresing the connectivity in tier 2 and tier 3 cities along with profession tie-ups with several e-commerce giants can increase the revenue as well as the reputation on connectivity across borders. We can work on optimizing the scanning time on both ends which is start scanning time and end scanning time so that the delivery time can be equated to the OSRM estimated delivery time.
- Revisit information fed to routing engine for trip planning. Check for discrepancies with transporters, if the routing engine is configured for optimum results.
- North, South and West Zones comidors have significant traffic of orders. But, we have a smaller presence in Central, Eastern and North-Eastern zone. However it would be difficult to conclude this, by looking at just 2 months data. It is worth investigating and increasing our presence in these regions.
- From state point of view, we have heavy traffic in Mahrashtra followed by Karnataka. This is a good indicator that we need to plan for resources on ground in these 2 states on priority. Especially, during festive seasons.

[ ]: