In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df=pd.read_csv("/Users/praneetcb/Documents/delhivery_data.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	sou
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND

5 rows × 24 columns

In [6]:
```
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
```

In [7]:
```
df.nunique()
```

Out[7]:
```
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
```

In [8]:
```
df.describe()
```

Out[8]:

	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	actual_time
count	144867.000000	144867.000000	144867.000000	144867.000000
mean	961.262986	232.926567	234.073372	416.927527
std	1037.012769	344.755577	344.990009	598.103621
min	20.000000	9.000000	9.000045	9.000000
25%	161.000000	22.000000	23.355874	51.000000
50%	449.000000	66.000000	66.126571	132.000000
75%	1634.000000	286.000000	286.708875	513.000000
max	7898.000000	1927.000000	1927.447705	4532.000000

In [16]:
```
# Converting columns to date_time object
df['od_start_time']=pd.to_datetime(df['od_start_time'])
df['od_end_time']=pd.to_datetime(df['od_end_time'])
```

In [20]:
```
# Removing null values and resetting index
df=df.dropna(how='any')
df=df.reset_index(drop=True)
```

In [21]:
```
df.isna().sum()
```

Out[21]:
```
data                                    0
trip_creation_time                      0
route_schedule_uuid                     0
route_type                              0
trip_uuid                               0
source_center                           0
source_name                             0
destination_center                      0
destination_name                        0
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
```

In [ ]:
```
#aggreagating data basis on trip_id, source_centre, Destination_cen
```

In [41]:
```
agg_columns = {
    'data' : 'first',
    'trip_creation_time' : 'first',
    'route_schedule_uuid' : 'first',
    'route_type' : 'first',
    'source_name' : 'first',
    'destination_name' : 'last',
    'start_scan_to_end_scan': 'first',
    'cutoff_timestamp': 'sum',
    'actual_distance_to_destination': 'last',
    'cutoff_factor': 'first',
    'segment_actual_time': 'sum',
    'segment_osrm_time': 'sum',
    'segment_osrm_distance': 'sum',
    'segment_factor': 'first',
    'actual_time': 'last',
    'osrm_time': 'last',
    'osrm_distance': 'last',
    'factor': 'sum',
    'od_start_time': 'first',
    'od_end_time': 'last',
    'is_cutoff': 'first',
    'cutoff_factor': 'first',
    'segment_factor': 'last'
}
grouped_data=df.groupby(['trip_uuid','source_center','destination_c
```

In [42]:
```
grouped_data
```

Out[42]:

data	trip_creation_time	route_schedule_uuid	route_type	source_name	
aining	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	Kanpur_Central_H_6 (Uttar Pradesh)	Gu
aining	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	Bhopal_Trnsport_H (Madhya Pradesh)	K
aining	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	Doddablpur_ChikaDPP_D (Karnataka)	Chik
aining	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	Tumkur_Veersagr_I (Karnataka)	Dodda
aining	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	Gurgaon_Bilaspur_HB (Haryana)	Chandi
...	...	...	...	...	
test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	Tirchchndr_Shnmgprm_D (Tamil Nadu)	Thisaya
test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	Peikulam_SriVnktpm_D (Tamil Nadu)	Tir
test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	Eral_Busstand_D (Tamil Nadu)	Tirchc
test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	Sandur_WrdN1DPP_D (Karnataka)	Bel
test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	Hospet (Karnataka)	Sar

In [ ]:
```
# Build some features to prepare the data for actual analysis. Extr
```

In [ ]:
```
# Destination Name: Split and extract features out of destination.
```

In [167]:
```
grouped_data['destination_city']=grouped_data['destination_name'].a
```

In [160]:
```
def name_place(x):

    # we will remove state
    x = x.split('(')[0]

    len_ = len(x.split('_'))

    if len_ >= 3:
        return x.split('_')[1]

    # small cities have same city and place name
    if len_ == 2:
        return x.split('_')[0]

    # now we need to deal with edge cases or imporper name conventi

    # if len(x.split('_')) == 2:

    return x.split(' ')[0]
```

In [168]:
```
grouped_data['destination_place']=grouped_data['destination_name'].
```

In [169]:
```
_data['destination_state'] = grouped_data['destination_name'].apply(
```

In [164]:
```
def name_code(x):
    # we will remove state
    x = x.split('(')[0]

    if len(x.split('_')) >= 3:
        return x.split('_')[-1]

    return 'none'
```

In [165]:
```
grouped_data['destination_code']=grouped_data['destination_name'].a
```

In [180]:
```
grouped_data[['destination_state','destination_city','destination_p
```

Out[180]:

	destination_state	destination_city	destination_place	destination_code
0	Haryana	Gurgaon	Bilaspur	HB
1	Uttar Pradesh	Kanpur	Central	6
2	Karnataka	Chikblapur	ShntiSgr	D
3	Karnataka	Doddablpur	ChikaDPP	D
4	Punjab	Chandigarh	Mehmdpur	H
...	...	...	...	...
26217	Tamil Nadu	Thisayanvilai	UdnkdiRD	D
26218	Tamil Nadu	Tirunelveli	VdkkuSrt	I
26219	Tamil Nadu	Tirchchndr	Shnmgprm	D
26220	Karnataka	Bellary	Bellary	none
26221	Karnataka	Sandur	WrdN1DPP	D

26222 rows × 4 columns

In [ ]:
```

```

In [ ]:
```
# Source Name: Split and extract features out of source. City_place
```

In [170]:
```
grouped_data['source_city']=grouped_data['source_name'].apply(lambd
```

In [172]:
```
grouped_data['source_place']=grouped_data['source_name'].apply(name
```

In [174]:
```
grouped_data['source_code']=grouped_data['source_name'].apply(name_
```

In [178]:
```
grouped_data['source_state'] = grouped_data['source_name'].apply(la
```

In [181]:
```
grouped_data[['source_state','source_city','source_place','source_c
```

Out[181]:

	source_state	source_city	source_place	source_code
0	Uttar Pradesh	Kanpur	Central	6
1	Madhya Pradesh	Bhopal	Trnsport	H
2	Karnataka	Doddablpur	ChikaDPP	D
3	Karnataka	Tumkur	Veersagr	I
4	Haryana	Gurgaon	Bilaspur	HB
...	...	...	...	...
26217	Tamil Nadu	Tirchchndr	Shnmgprm	D
26218	Tamil Nadu	Peikulam	SriVnktpm	D
26219	Tamil Nadu	Eral	Busstand	D
26220	Karnataka	Sandur	WrdN1DPP	D
26221	Karnataka	Hospet (Karnataka)	Hospet	none

26222 rows × 4 columns

In [ ]:
```
# Trip_creation_time: Extract features like month, year and day etc
```

In [183]:
```
grouped_data['trip_creation_time'] = pd.to_datetime(grouped_data['t

grouped_data['trip_year'] = grouped_data['trip_creation_time'].dt.y
grouped_data['trip_month'] = grouped_data['trip_creation_time'].dt.
grouped_data['trip_hour'] = grouped_data['trip_creation_time'].dt.h
grouped_data['trip_day'] = grouped_data['trip_creation_time'].dt.da
grouped_data['trip_week'] = grouped_data['trip_creation_time'].dt.i
grouped_data['trip_dayofweek'] = grouped_data['trip_creation_time']
```

In [185]:
```
grouped_data[['trip_year','trip_month','trip_hour','trip_day','trip
```

Out[185]:

	trip_year	trip_month	trip_hour	trip_day	trip_week	trip_dayofweek
0	2018.0	9.0	0.0	12.0	37	2.0
1	2018.0	9.0	0.0	12.0	37	2.0
2	2018.0	9.0	0.0	12.0	37	2.0
3	2018.0	9.0	0.0	12.0	37	2.0
4	2018.0	9.0	0.0	12.0	37	2.0
...	...	...	...	...	...	...
26217	NaN	NaN	NaN	NaN	<NA>	NaN
26218	NaN	NaN	NaN	NaN	<NA>	NaN
26219	NaN	NaN	NaN	NaN	<NA>	NaN
26220	NaN	NaN	NaN	NaN	<NA>	NaN
26221	NaN	NaN	NaN	NaN	<NA>	NaN

26222 rows × 6 columns

In [ ]:
```

```

In [ ]:
```
# Calculate time taken between od_start_time and od_end_time and cr
```

In [43]:
```
grouped_data['od_time_diff_hour'] = (grouped_data['od_end_time'] -
grouped_data['od_time_diff_hour']
```

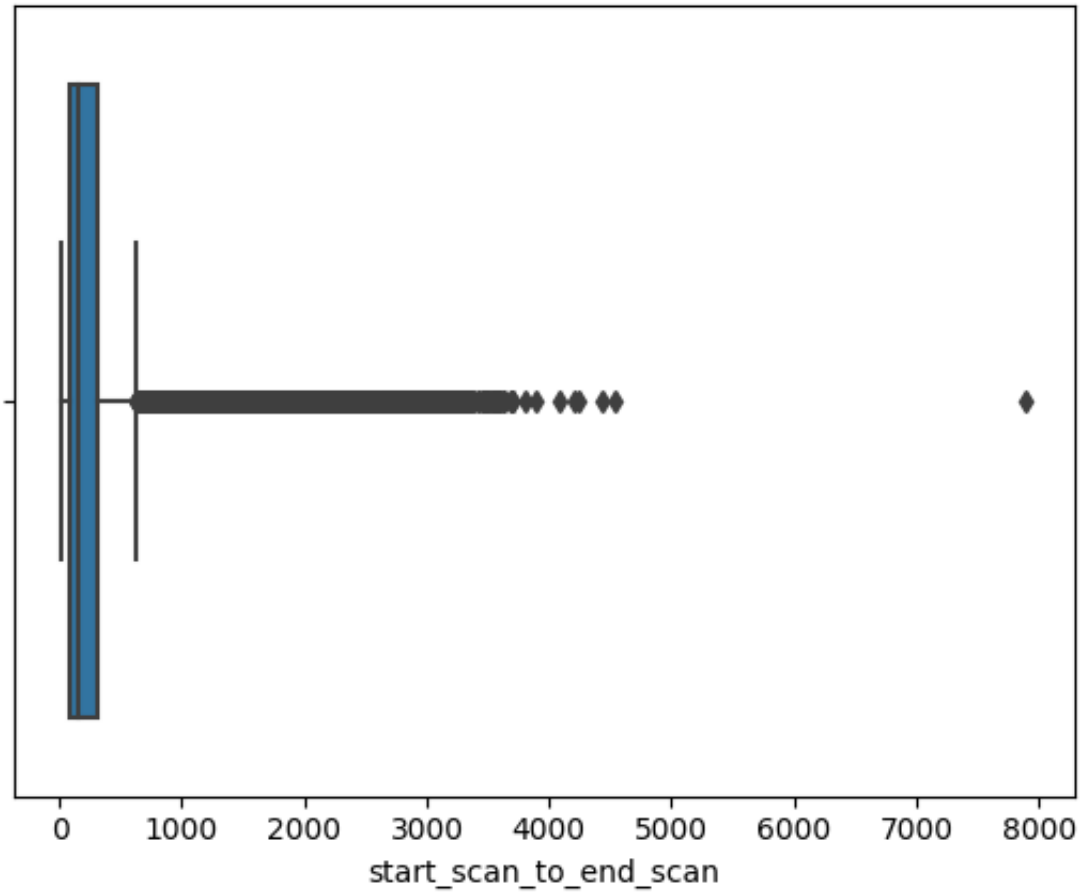
Out[43]:
```
0        1260.604421
1         999.505379
2          58.832388
3         122.779486
4         834.638929
              ...
26217      62.115193
26218      91.087797
26219      44.174403
26220     287.474007
26221      66.933565
Name: od_time_diff_hour, Length: 26222, dtype: float64
```

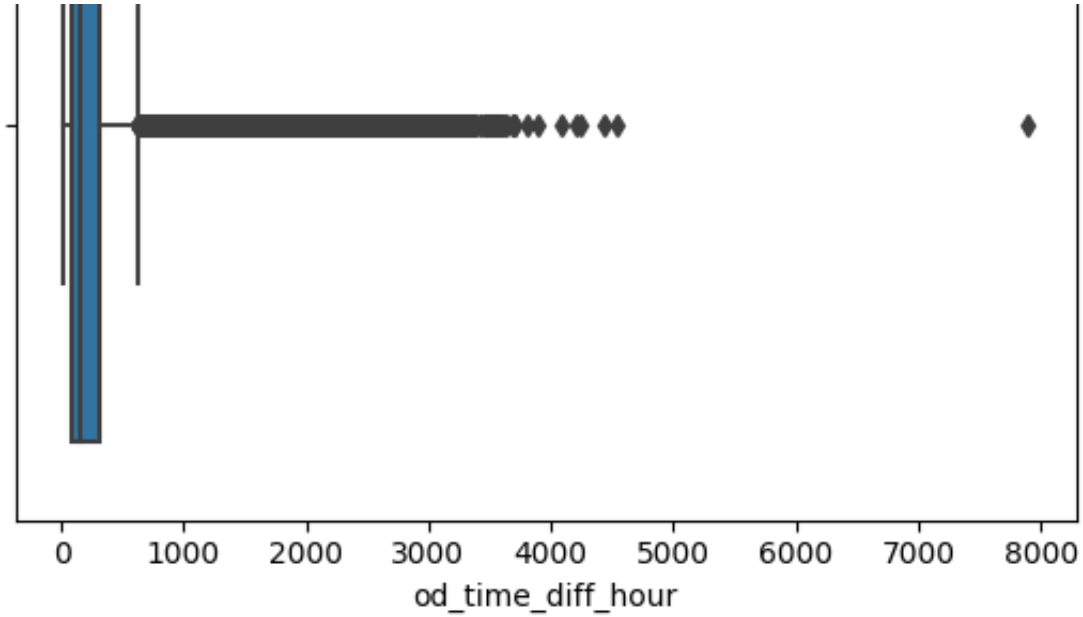
In [44]:
```
grouped_data['start_scan_to_end_scan']
```

Out[44]:
```
0        1260.0
1         999.0
2          58.0
3         122.0
4         834.0
          ...
26217      62.0
26218      91.0
26219      44.0
26220     287.0
26221      66.0
Name: start_scan_to_end_scan, Length: 26222, dtype: float64
```

In [ ]:
```
#od_time_diff_hour is matching with start_scan_to_end_scan
```

In [55]:
```
sns.boxplot(data=grouped_data, x='start_scan_to_end_scan')
plt.show()
sns.boxplot(data=grouped_data, x='od_time_diff_hour')
plt.show()
```





In [ ]:
```
#In-depth analysis and hypothesis testing
```

In [ ]:
```
#Creating aggregated values for all the tests - these are the value
#and taking sum of the values that are required for test.
```

In [60]:
```
trip_dict = {

    'data' : 'first',
    'trip_creation_time' : 'first',
    'route_schedule_uuid' : 'first',
    'route_type' : 'first',
    'trip_uuid' : 'first',

    'source_center' : 'first',
    'source_name' : 'first',

    'destination_center' : 'last',
    'destination_name' : 'last',

    'start_scan_to_end_scan' : 'sum',
    'od_time_diff_hour' : 'sum',

    'actual_distance_to_destination' : 'sum',
    'actual_time' : 'sum',
    'osrm_time' : 'sum',
    'osrm_distance' : 'sum',

    'segment_actual_time' : 'sum',
    'segment_osrm_distance' : 'sum',
    'segment_osrm_time' : 'sum',

}
```

In [61]:
```
trip=grouped_data.groupby('trip_uuid').agg(trip_dict).reset_index(d
```

In [62]:
```
trip
```

Out[62]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid
0	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	trip-153671041653548748
1	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	trip-153671042288605164
2	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	trip-153671043369099517
3	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	trip-153671046011330457
4	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	trip-153671052974046625
...	...	...	...	...	...
14782	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	trip-153861095625827784
14783	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	trip-153861104386292051
14784	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	trip-153861106442901555
14785	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	trip-153861115439069069
14786	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	trip-153861118270144424

14787 rows × 18 columns

In [ ]:
```
# Visualize the relationship between two numerical values, such as
```

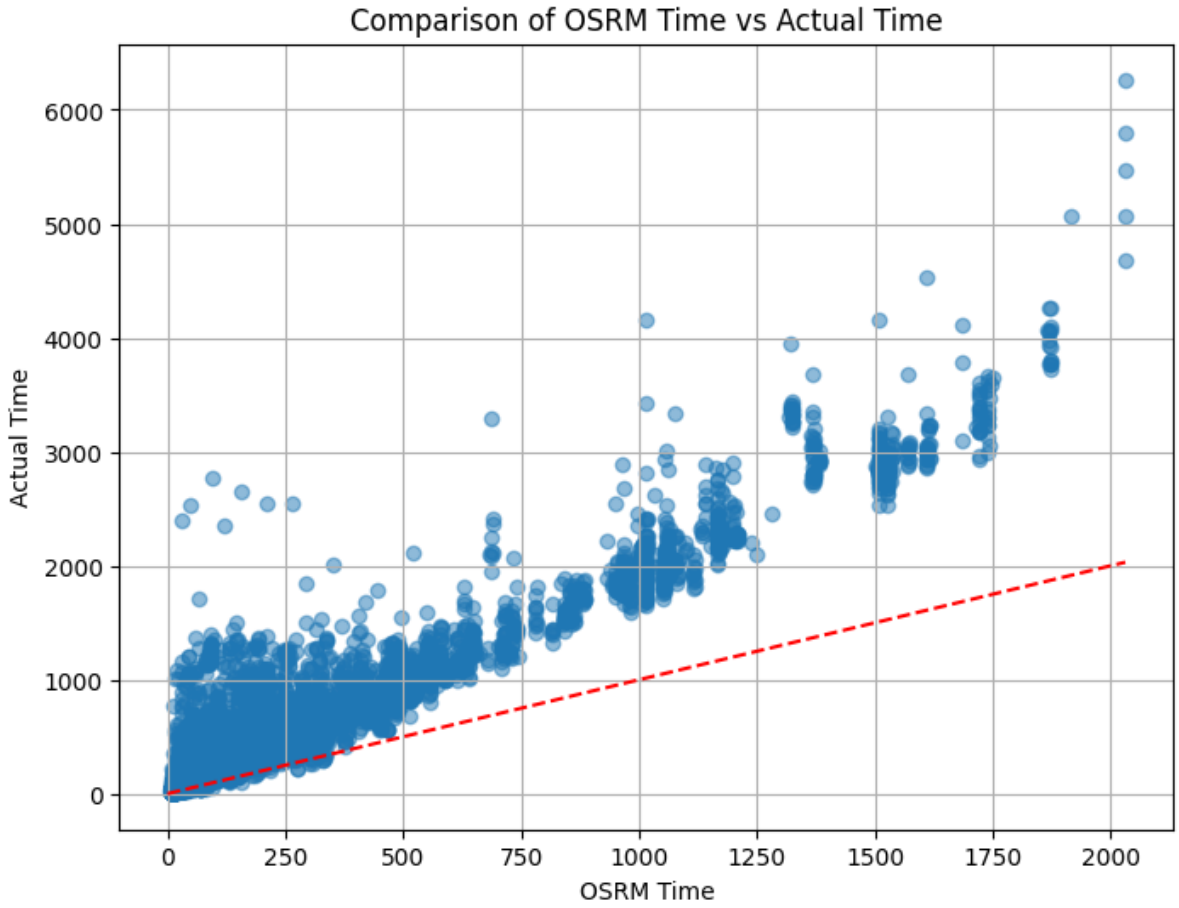
In [64]:
```
trip[['actual_time', 'osrm_time']]
```

Out[64]:

	actual_time	osrm_time
0	1562.0	717.0
1	143.0	68.0
2	3347.0	1740.0
3	59.0	15.0
4	341.0	117.0
...	...	...
14782	83.0	62.0
14783	21.0	12.0
14784	282.0	48.0
14785	264.0	179.0
14786	275.0	68.0

14787 rows × 2 columns

In [70]:
```
plt.figure(figsize=(8, 6))
plt.scatter(trip['osrm_time'], trip['actual_time'], alpha=0.5)
plt.plot(np.arange(0, max(trip['osrm_time'])), np.arange(0, max(tri
plt.xlabel('OSRM Time')
plt.ylabel('Actual Time')
plt.title('Comparison of OSRM Time vs Actual Time')
plt.grid(True)
plt.show()
```



In [78]:
```
from scipy.stats import ttest_rel, ttest_ind
```

In [81]:
```
# Hypothesis testing using T-test-ind for actual time and segment a
```

In [82]:
```
trip[['actual_time', 'segment_actual_time']]
```

Out[82]:

	actual_time	segment_actual_time
0	1562.0	1548.0
1	143.0	141.0
2	3347.0	3308.0
3	59.0	59.0
4	341.0	340.0
...	...	...
14782	83.0	82.0
14783	21.0	21.0
14784	282.0	281.0
14785	264.0	258.0
14786	275.0	274.0

14787 rows × 2 columns

In [83]:
```
t_test, p_val = ttest_ind(trip['actual_time'],trip['segment_actual_
```

In [84]:
```
# Print t-test results
print("Paired t-test results:")
print("T-statistic:", t_test)
print("P-value:", p_val)

# Interpret t-test results
alpha = 0.05
if p_val < alpha:
    print("Reject null hypothesis. There is a significant differenc
else:
    print("Fail to reject null hypothesis. There is no significant
```

```
Paired t-test results:
T-statistic: 0.499475764573994
P-value: 0.6174479719707524
Fail to reject null hypothesis. There is no significant difference
between actual_time and osrm_time.
```

In [ ]:

In [ ]:
```
# Visual analysis between osrm distance and segment osrm distance v
```

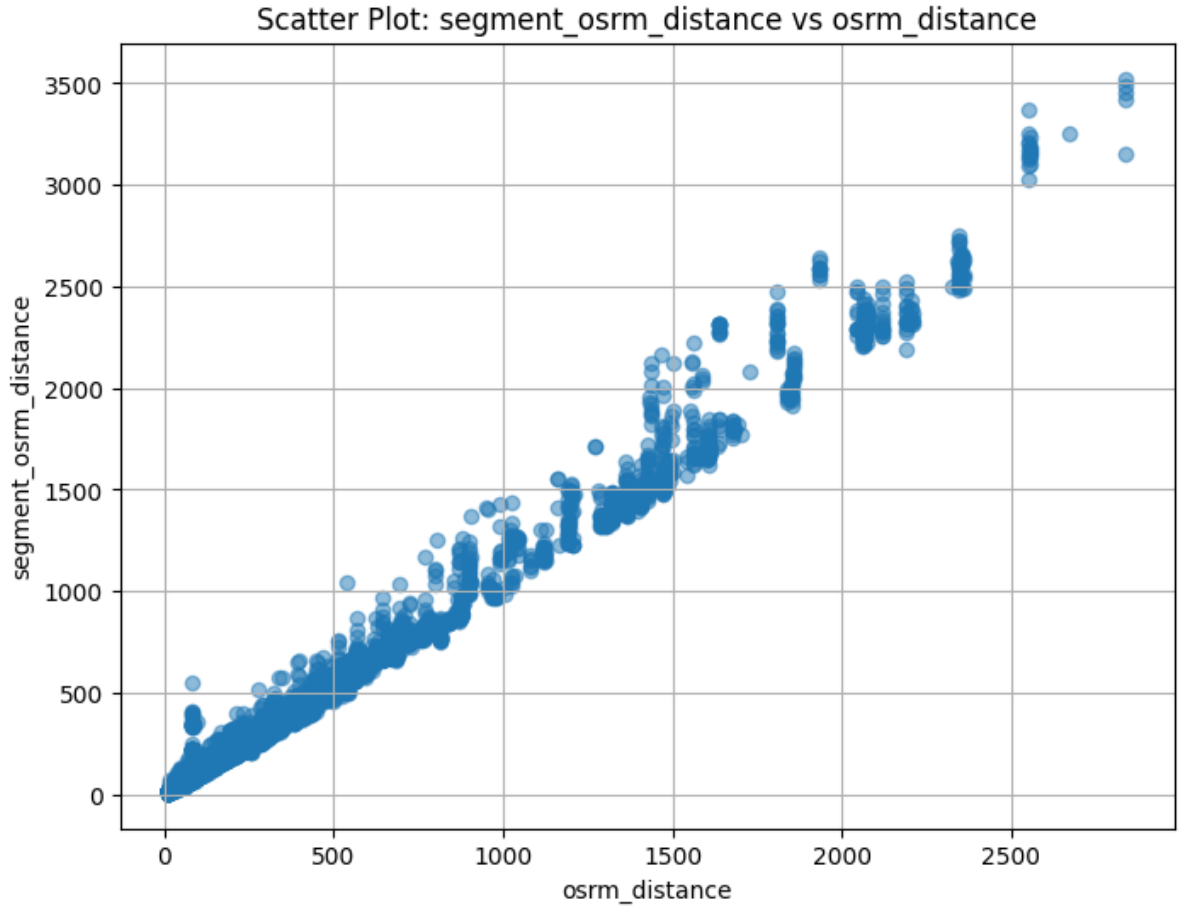
In [89]:
```
trip[['osrm_distance','segment_osrm_distance']]
```

Out[89]:

	osrm_distance	segment_osrm_distance
0	991.3523	1320.4733
1	85.1110	84.1894
2	2354.0665	2545.2678
3	19.6800	19.8766
4	146.7918	146.7919
...	...	...
14782	73.4630	64.8551
14783	16.0882	16.0883
14784	58.9037	104.8866
14785	171.1103	223.5324
14786	80.5787	80.5787

14787 rows × 2 columns

In [91]:
```
x_column = 'osrm_distance'
y_column = 'segment_osrm_distance'

# Create a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(trip[x_column], trip[y_column], alpha=0.5)
plt.xlabel(x_column)
plt.ylabel(y_column)
plt.title(f'Scatter Plot: {y_column} vs {x_column}')
plt.grid(True)
plt.show()
```



In [ ]:
```
# Visual analysis between osrm time and segment osrm time value
```

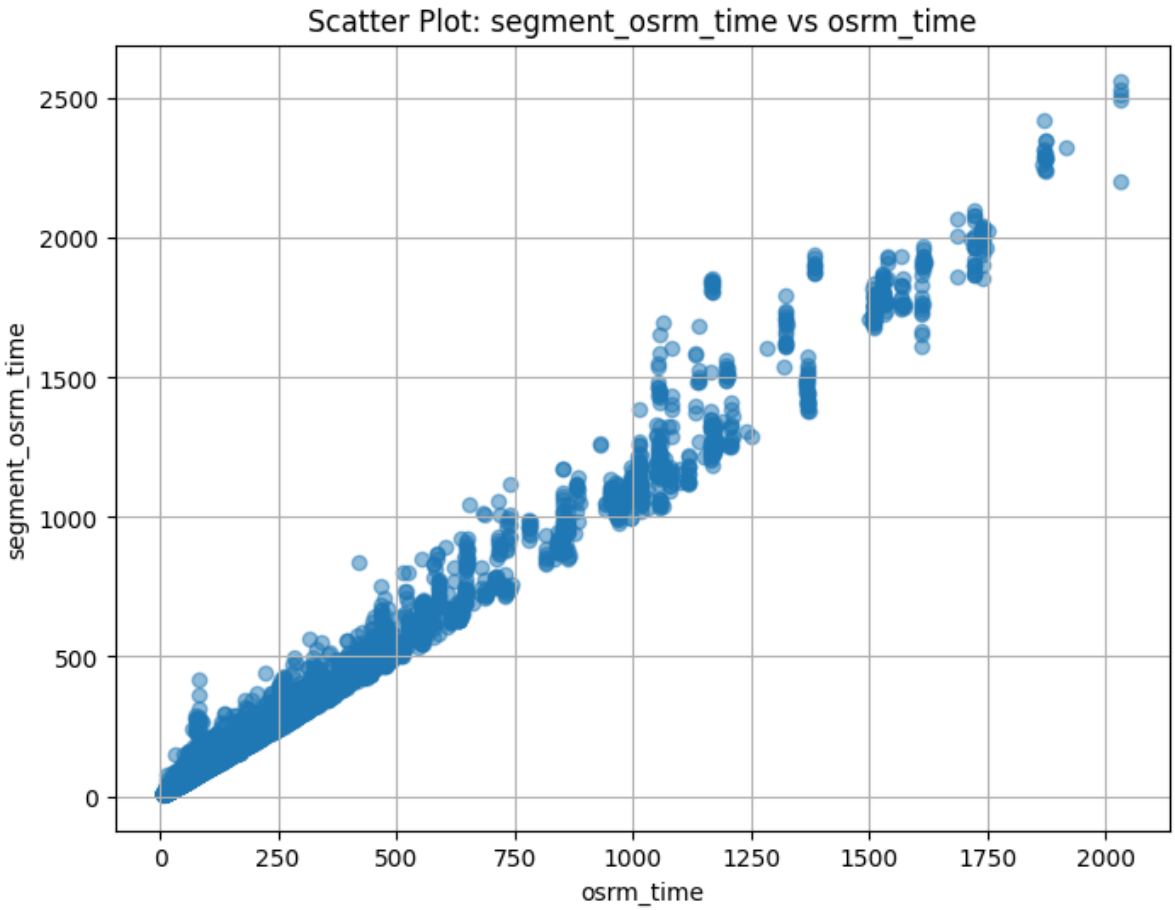
In [92]:
```
trip[['osrm_time','segment_osrm_time']]
```

Out[92]:

	osrm_time	segment_osrm_time
0	717.0	1008.0
1	68.0	65.0
2	1740.0	1941.0
3	15.0	16.0
4	117.0	115.0
...	...	...
14782	62.0	62.0
14783	12.0	11.0
14784	48.0	88.0
14785	179.0	221.0
14786	68.0	67.0

14787 rows × 2 columns

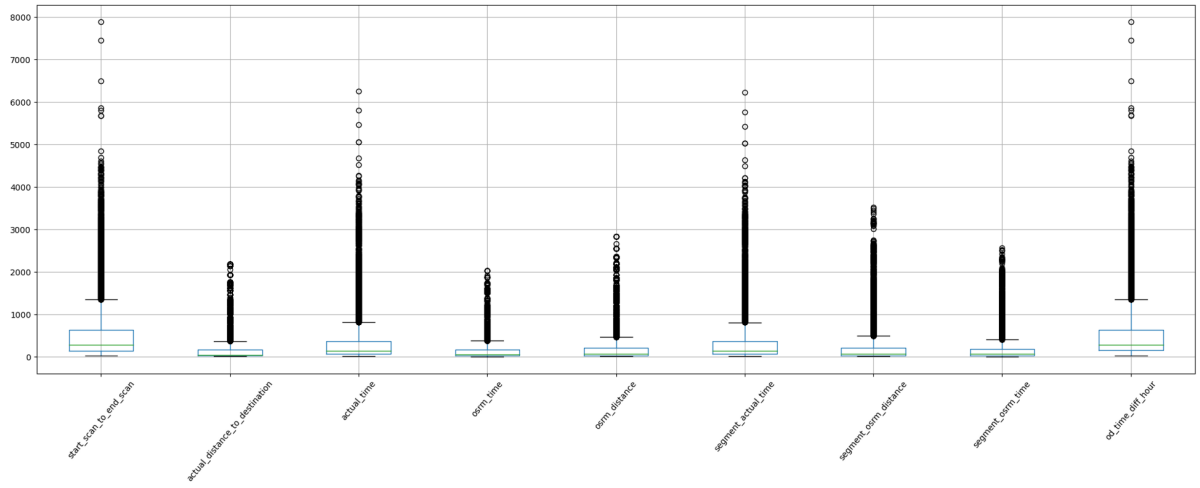
In [93]:
```
x_column = 'osrm_time'
y_column = 'segment_osrm_time'

# Create a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(trip[x_column], trip[y_column], alpha=0.5)
plt.xlabel(x_column)
plt.ylabel(y_column)
plt.title(f'Scatter Plot: {y_column} vs {x_column}')
plt.grid(True)
plt.show()
```



In [ ]:
```
# Outliers in the numerical variables (you might find outliers in a
```

In [95]:
```
num_cols = ['start_scan_to_end_scan','actual_distance_to_destinatio
            'osrm_distance','segment_actual_time','segment_osrm_dis
           'segment_osrm_time', 'od_time_diff_hour']
```

In [99]:
```
trip[num_cols].boxplot(rot=50, figsize=(25,8))
plt.show()
```



In [ ]:
```
# Handle the outliers using the IQR method.
```

In [102]:
```
for col in num_cols:
    Q1 = trip[col].quantile(0.25)
    Q3 = trip[col].quantile(0.75)
    IQR = Q3 - Q1

    upper_bound = Q3 + 1.5 * IQR
    lower_bound = Q1 - 1.5 * IQR

    trip[col]=trip[col].apply(lambda x: x if lower_bound <= x <= up
```

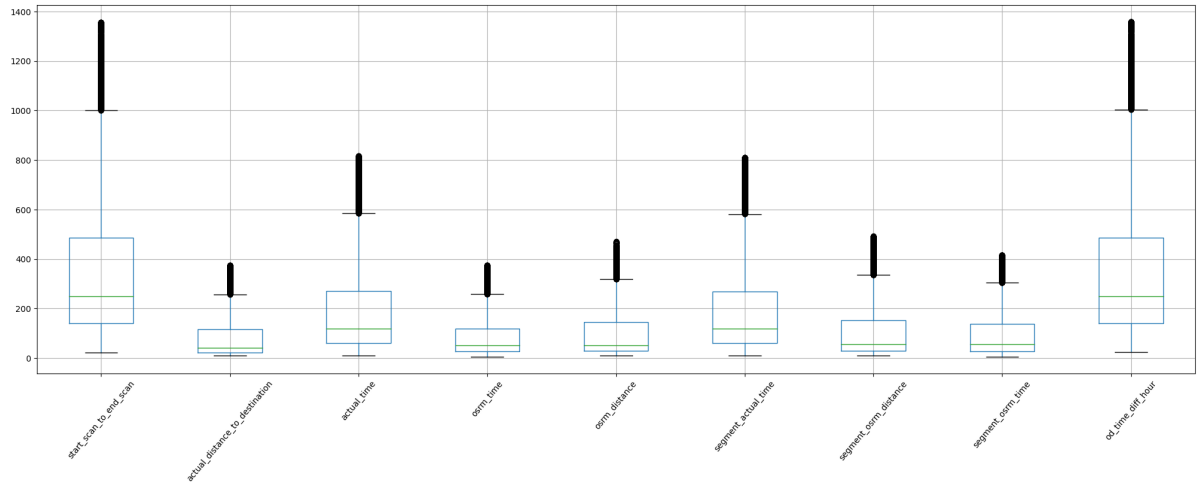
In [103]:
```
trip
```

Out[103]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid
0	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	trip-153671041653548748
1	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	trip-153671042288605164
2	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	trip-153671043369099517
3	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	trip-153671046011330457
4	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	trip-153671052974046625
...	...	...	...	...	...
14782	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	trip-153861095625827784
14783	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	trip-153861104386292051
14784	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	trip-153861106442901555
14785	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	trip-153861115439069069
14786	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	trip-153861118270144424

14787 rows × 18 columns

In [104]:
```
trip[num_cols].boxplot(rot=50, figsize=(25,8))
plt.show()
```



In [105]:
```
trip.isna().sum()
```

Out[105]:
```
data                                0
trip_creation_time                  0
route_schedule_uuid                 0
route_type                          0
trip_uuid                           0
source_center                       0
source_name                         0
destination_center                  0
destination_name                    0
start_scan_to_end_scan           1282
od_time_diff_hour                1275
actual_distance_to_destination   1452
actual_time                      1646
osrm_time                        1506
osrm_distance                    1522
segment_actual_time              1644
segment_osrm_distance            1550
segment_osrm_time                1485
dtype: int64
```

In [ ]:
```
# Do one-hot encoding of categorical variables (like route_type)
```

In [106]:
```
trip['route_type'].value_counts()
```

Out[106]:
```
route_type
Carting    8906
FTL        5881
Name: count, dtype: int64
```

In [107]:
```
trip['route_type']=trip['route_type'].map({'FTL':0,'Carting':1}) #
```

In [108]:
```
trip['route_type'].value_counts()
```

Out[108]:
```
route_type
1    8906
0    5881
Name: count, dtype: int64
```

In [ ]:
```

```

In [ ]:
```
# Normalize/ Standardize the numerical features using MinMaxScaler
```

In [111]:
```
pip install -U scikit-learn
```

```
Collecting scikit-learn
  Downloading scikit_learn-1.3.0-cp39-cp39-macosx_10_9_x86_64.whl
(10.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.2/10.2 MB 1.5 MB/
s eta 0:00:0000:010:01
Requirement already satisfied: scipy>=1.5.0 in ./opt/anaconda3/lib
/python3.9/site-packages (from scikit-learn) (1.10.1)
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.2.0-py3-none-any.whl (15 kB)
Requirement already satisfied: numpy>=1.17.3 in ./opt/anaconda3/li
b/python3.9/site-packages (from scikit-learn) (1.24.2)
Collecting joblib>=1.1.1
  Downloading joblib-1.3.2-py3-none-any.whl (302 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 302.2/302.2 kB 1.7 MB
/s eta 0:00:00a 0:00:01
Installing collected packages: threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.3.2 scikit-learn-1.3.0 threadpoolc
tl-3.2.0
Note: you may need to restart the kernel to use updated packages.
```

In [112]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [118]:
```
scaler=MinMaxScaler()
scaled_data=scaler.fit_transform(trip[num_cols])
scaled_data=pd.DataFrame(scaled_data, columns=trip[num_cols].column
```

In [122]:
```
scaled_data.dropna(inplace=True)
```

In [123]:
```
scaled_data
```

Out[123]:

	start_scan_to_end_scan	actual_distance_to_destination	actual_time	osrm_time	osrn
1	0.117779	0.175460	0.165842	0.167568	
3	0.057764	0.022342	0.061881	0.024324	
4	0.520630	0.323794	0.410891	0.300000	
5	0.124531	0.042631	0.064356	0.045946	
6	0.056264	0.000268	0.018564	0.018919	
...	...	...	...	...	
14782	0.175544	0.133294	0.091584	0.151351	
14783	0.027757	0.017800	0.014851	0.016216	
14784	0.298575	0.081142	0.337871	0.113514	
14785	0.243061	0.343682	0.315594	0.467568	
14786	0.247562	0.156036	0.329208	0.167568	

12723 rows × 9 columns

In [124]:
```
scaled_data.describe()
```

Out[124]:

	start_scan_to_end_scan	actual_distance_to_destination	actual_time	osrm_time
count	12723.000000	12723.000000	12723.000000	12723.000000
mean	0.222940	0.173084	0.208481	0.195785
std	0.191715	0.197017	0.195731	0.195496
min	0.000000	0.000000	0.000000	0.000000
25%	0.084771	0.033879	0.064356	0.056757
50%	0.157539	0.080706	0.129950	0.118919
75%	0.300075	0.253333	0.299505	0.278378
max	0.999250	0.996260	0.997525	1.000000

Recommendation Examples:
There is a significant difference between OSRM and actual parameters.

There is a need to:
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.

In [ ]:

In [ ]: