# Bussiness Case:

**Delhivery is the largest and fastest-growing fully integrated player in India by revenue in Fiscal 2021. They aim to build the operating system for commerce, through a combination of world-class infrastructure, logistics operations of the highest quality, and cutting-edge engineering and technology capabilities.The Data team builds intelligence and capabilities using this data that helps them to widen the gap between the quality, efficiency, and profitability of their business versus their competitors.**

In [1]:

```
#importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import datetime as dt
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from scipy import stats
```

In [21]:

```
#importing file
df=pd.read_csv('Delhivery.csv')
```

In [3]:

```
df.shape
```

Out[3]:

```
(144867, 24)
```

In [4]:

```
df.head(5)
```

Out[4]:

| | data | trip_creation_time | route_schedule_uuid | route_type | trip_uuid | source_center | source_name |
|---|---|---|---|---|---|---|---|
| 0 | training | 9/20/2018 2:35:36 AM | thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3... | Carting | trip-153741093647649320 | IND388121AAA | Anand_VUNagar_DC (Gujarat) |
| 1 | training | 9/20/2018 2:35:36 AM | thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3... | Carting | trip-153741093647649320 | IND388121AAA | Anand_VUNagar_DC (Gujarat) |
| 2 | training | 9/20/2018 2:35:36 AM | thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3... | Carting | trip-153741093647649320 | IND388121AAA | Anand_VUNagar_DC (Gujarat) |
| 3 | training | 9/20/2018 2:35:36 AM | thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3... | Carting | trip-153741093647649320 | IND388121AAA | Anand_VUNagar_DC (Gujarat) |
| 4 | training | 9/20/2018 2:35:36 AM | thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3... | Carting | trip-153741093647649320 | IND388121AAA | Anand_VUNagar_DC (Gujarat) |

5 rows × 24 columns

In [5]:

```
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  int64  
 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  int64  
 17  osrm_time                       144867 non-null  int64  
 18  osrm_distance                   144867 non-null  float64
 19  factor                          144867 non-null  float64
 20  segment_actual_time             144867 non-null  int64  
 21  segment_osrm_time               144867 non-null  int64  
 22  segment_osrm_distance           144867 non-null  float64
 23  segment_factor                  144867 non-null  float64
dtypes: bool(1), float64(5), int64(6), object(12)
memory usage: 25.6+ MB
```

In [6]:

```
df.describe()
```

Out[6]:

| | start_scan_to_end_scan | cutoff_factor | actual_distance_to_destination | actual_time | osrm_time | osrm_distance | 1 |
|---|---|---|---|---|---|---|---|
| **count** | 144867.000000 | 144867.000000 | 144867.000000 | 144867.000000 | 144867.000000 | 144867.000000 | |
| **mean** | 961.262986 | 232.926567 | 234.073372 | 416.927527 | 213.868272 | 284.771297 | |
| **std** | 1037.012769 | 344.755577 | 344.990009 | 598.103621 | 308.011085 | 421.119294 | |
| **min** | 20.000000 | 9.000000 | 9.000045 | 9.000000 | 6.000000 | 9.008200 | |
| **25%** | 161.000000 | 22.000000 | 23.355874 | 51.000000 | 27.000000 | 29.914700 | |
| **50%** | 449.000000 | 66.000000 | 66.126571 | 132.000000 | 64.000000 | 78.525800 | |
| **75%** | 1634.000000 | 286.000000 | 286.708875 | 513.000000 | 257.000000 | 343.193250 | |
| **max** | 7898.000000 | 1927.000000 | 1927.447705 | 4532.000000 | 1686.000000 | 2326.199100 | |

In [8]:

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

Out[8]:

| | data | trip_creation_time | route_schedule_uuid | route_type | trip_uuid | source_center | source |
|---|---|---|---|---|---|---|---|
| **count** | 144867 | 144867 | 144867 | 144867 | 144867 | 144867 | |
| **unique** | 2 | 14754 | 1504 | 2 | 14817 | 1508 | |
| **top** | training | 9/13/2018 7:44:52 PM | thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069f... | FTL | trip-153811219535896559 | IND000000ACB | Gurgaon_Bilasp (Ha |

| | data | trip_creation_time | route_schedule_uuid | route_type | trip_uuid | source_center | source |
|---|---|---|---|---|---|---|---|
| freq | 104858 | 122 | 1812 | 99660 | 101 | 23347 | |

In [20]:

```
#Checking Null values
df.isna().sum()
```

Out[20]:

```
data                                 0
trip_creation_time                   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
actual_distance_to_destination       0
actual_time                          0
osrm_time                            0
osrm_distance                        0
segment_actual_time                  0
segment_osrm_time                    0
segment_osrm_distance                0
dtype: int64
```

In [10]:

```
#checking duplicates
df.duplicated().sum()
```

Out[10]:

```
0
```

In [22]:

```
#removing null values
df = df.dropna()
```

## Cleaning and aggregating data to get useful features out of raw fields

In [23]:

```
#copying Data Frame
df1=df
```

In [24]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 144316 entries, 0 to 144866
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype 
---  ------                          --------------   ----- 
 0   data                            144316 non-null  object
 1   trip_creation_time              144316 non-null  object
 2   route_schedule_uuid             144316 non-null  object
 3   route_type                      144316 non-null  object
 4   trip_uuid                       144316 non-null  object
 5   source_center                   144316 non-null  object
 6   source_name                     144316 non-null  object
 7   destination_center              144316 non-null  object
 8   destination_name                144316 non-null  object
```

```
 9   od_start_time                     144316 non-null  object
 10  od_end_time                       144316 non-null  object
 11  start_scan_to_end_scan            144316 non-null  int64
 12  is_cutoff                         144316 non-null  bool
 13  cutoff_factor                     144316 non-null  int64
 14  cutoff_timestamp                  144316 non-null  object
 15  actual_distance_to_destination    144316 non-null  float64
 16  actual_time                       144316 non-null  int64
 17  osrm_time                         144316 non-null  int64
 18  osrm_distance                     144316 non-null  float64
 19  factor                            144316 non-null  float64
 20  segment_actual_time               144316 non-null  int64
 21  segment_osrm_time                 144316 non-null  int64
 22  segment_osrm_distance             144316 non-null  float64
 23  segment_factor                    144316 non-null  float64
dtypes: bool(1), float64(5), int64(6), object(12)
memory usage: 26.6+ MB
```

In [25]:

```
#dropping unnessary columns
df1.drop(['route_schedule_uuid','is_cutoff','cutoff_factor','cutoff_timestamp','factor','
segment_factor'], axis=1, inplace=True)
```

In [26]:

```
#aggregating data
agg_func_selection = {'data': 'first',
                      'trip_creation_time':'first',
                      'route_type':'first',

'source_center':'first',
                      'source_name':'first',
                      'destination_center':'last',
                      'destination_name':'last',
 'od_start_time':'first',
                      'od_end_time':'last',
                      'start_scan_to_end_scan':'last',
                      'actual_distance_to_destination':'last',
                      'actual_time':'last',
                      'osrm_time':'last',
                      'osrm_distance':'last',
                      'segment_actual_time':'sum',
 'segment_osrm_time':'sum',
 'segment_osrm_distance':'sum'}
```

In [27]:

```
df2=pd.DataFrame(df1.groupby(['trip_uuid']).agg(agg_func_selection))
```

In [28]:

```
df2=df2.reset_index()
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14787 entries, 0 to 14786
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   trip_uuid                       14787 non-null  object
 1   data                            14787 non-null  object
 2   trip_creation_time              14787 non-null  object
 3   route_type                      14787 non-null  object
 4   source_center                   14787 non-null  object
 5   source_name                     14787 non-null  object
 6   destination_center              14787 non-null  object
 7   destination_name                14787 non-null  object
 8   od_start_time                   14787 non-null  object
 9   od_end_time                     14787 non-null  object
 10  start_scan_to_end_scan          14787 non-null  int64
 11  actual_distance_to_destination  14787 non-null  float64
```

```
 12  actual_time                   14787 non-null  int64
 13  osrm_time                     14787 non-null  int64
 14  osrm_distance                 14787 non-null  float64
 15  segment_actual_time           14787 non-null  int64
 16  segment_osrm_time             14787 non-null  int64
 17  segment_osrm_distance         14787 non-null  float64
dtypes: float64(3), int64(5), object(10)
memory usage: 2.0+ MB
```

In [29]:

```
df2['data'].value_counts(normalize='True')
```

Out[29]:

```
training    0.719889
test        0.280111
Name: data, dtype: float64
```

In [30]:

```
df2['route_type'].value_counts(normalize=True)
```

Out[30]:

```
Carting    0.602286
FTL        0.397714
Name: route_type, dtype: float64
```

In [31]:

```
#top 5 source city names
df2['source_name'].value_counts().head(5)
```

Out[31]:

```
Gurgaon_Bilaspur_HB (Haryana)         937
Bhiwandi_Mankoli_HB (Maharashtra)     811
Bangalore_Nelmngla_H (Karnataka)      731
Bengaluru_Bomsndra_HB (Karnataka)     426
Chandigarh_Mehmdpur_H (Punjab)        370
Name: source_name, dtype: int64
```

In [32]:

```
#top 10 destination city names
df2['destination_name'].value_counts().head(5)
```

Out[32]:

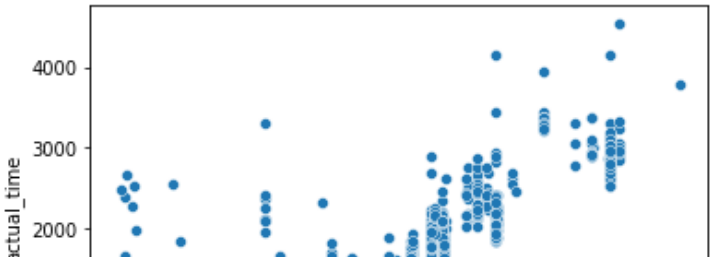
```
Gurgaon_Bilaspur_HB (Haryana)         813
Bangalore_Nelmngla_H (Karnataka)      628
Bhiwandi_Mankoli_HB (Maharashtra)     573
Chandigarh_Mehmdpur_H (Punjab)        431
Hyderabad_Shamshbd_H (Telangana)      400
Name: destination_name, dtype: int64
```

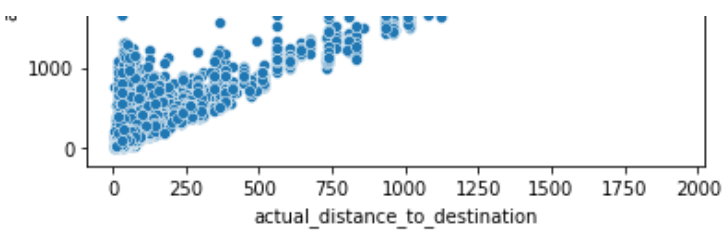
In [33]:

```
sns.scatterplot(df2['actual_distance_to_destination'],df2['actual_time'])
```

Out[33]:

```
<AxesSubplot:xlabel='actual_distance_to_destination', ylabel='actual_time'>
```






In [34]:

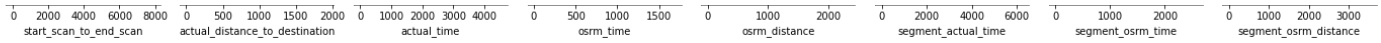
```
numerical=['start_scan_to_end_scan',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'segment_actual_time',
 'segment_osrm_time',
 'segment_osrm_distance']
```

In [35]:

```
sns.pairplot(df2[numerical])
```

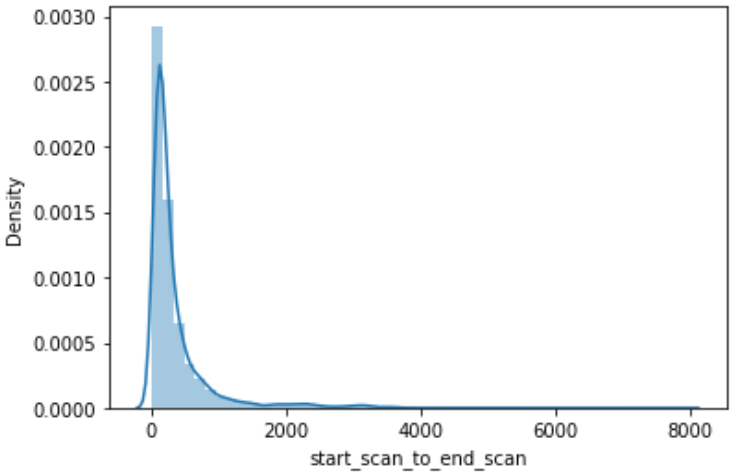
Out[35]:

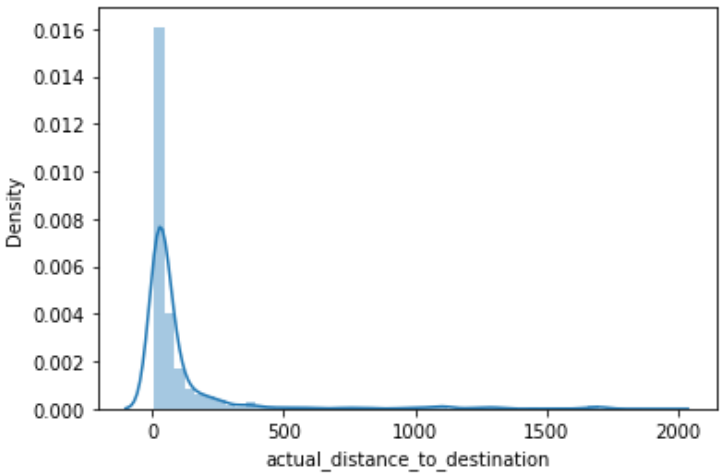
```
<seaborn.axisgrid.PairGrid at 0x1809107cac0>
```

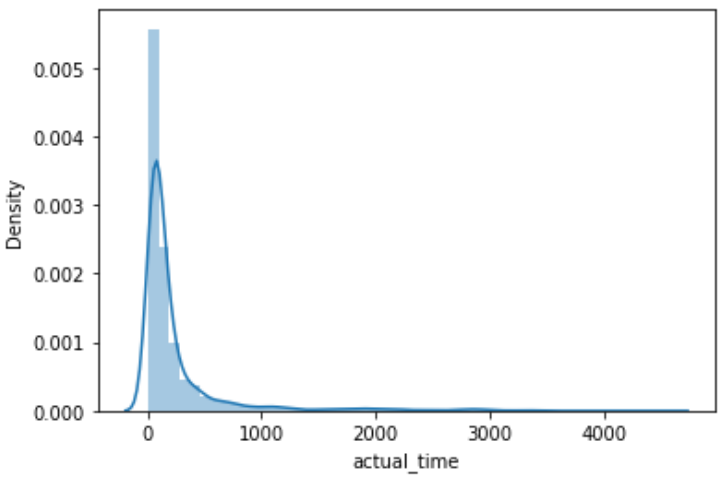


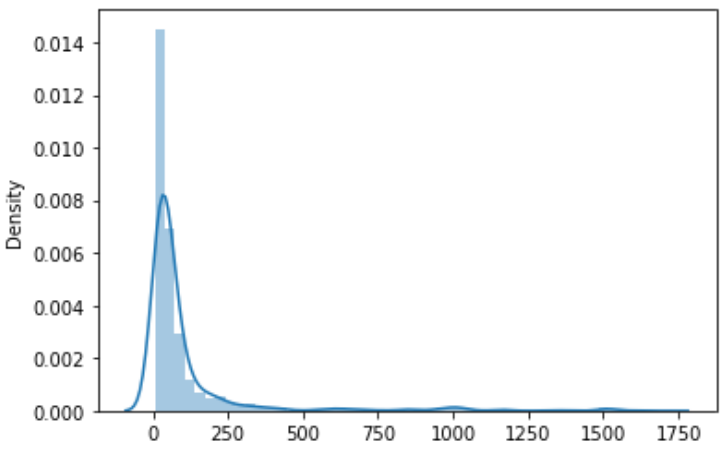

In [36]:

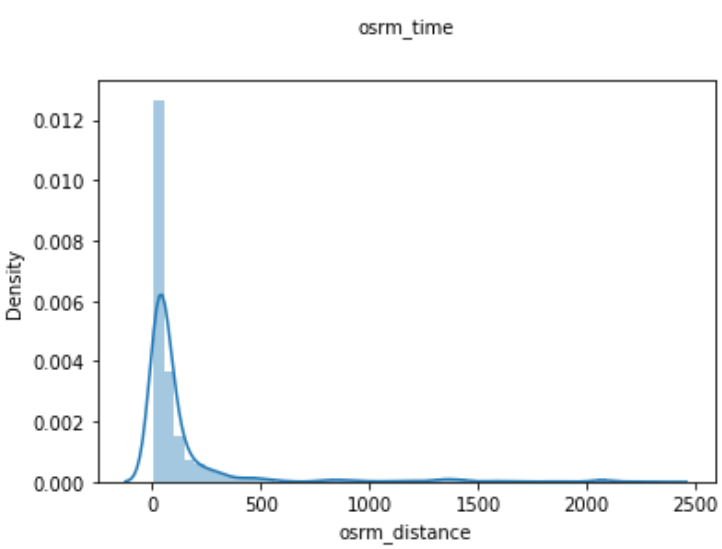
```
for i in numerical:
    plt.figure()
    sns.distplot(df2[i])
```

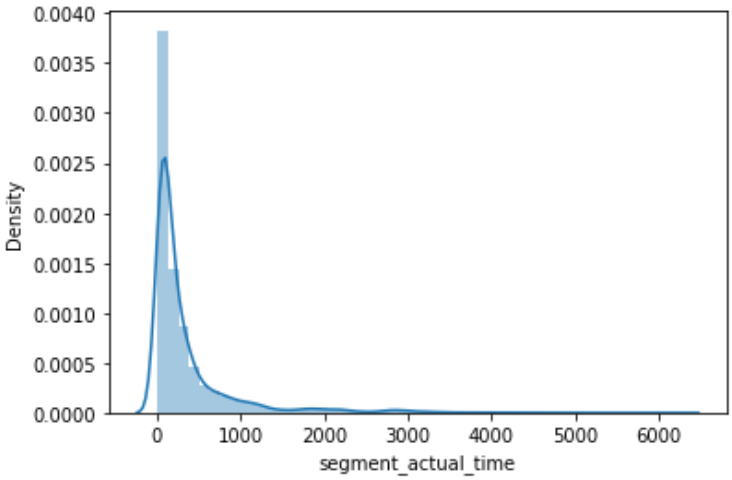


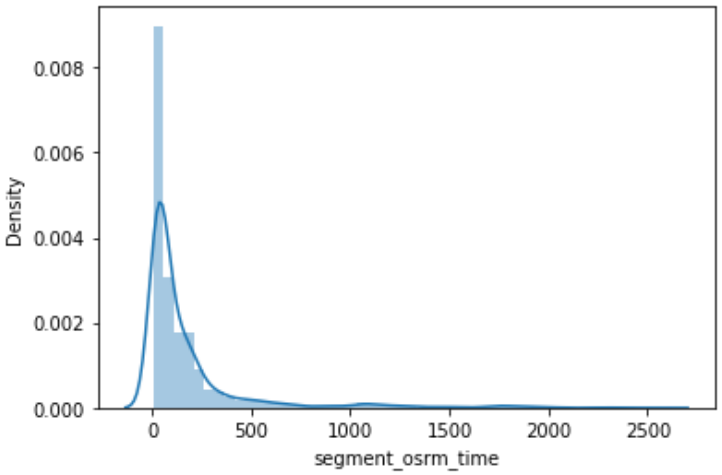



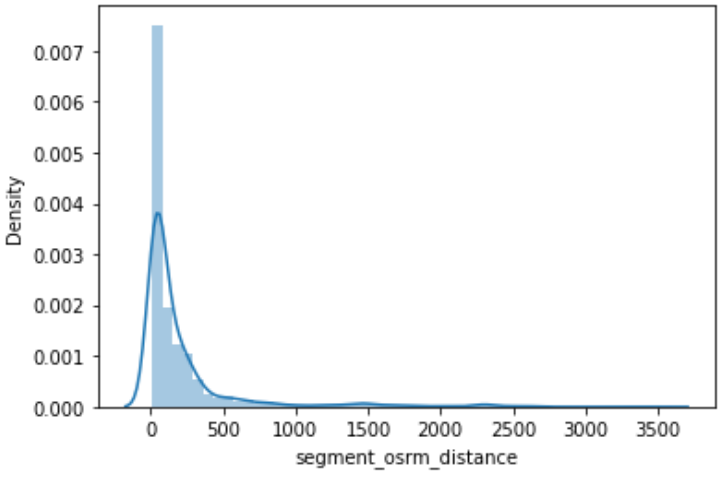





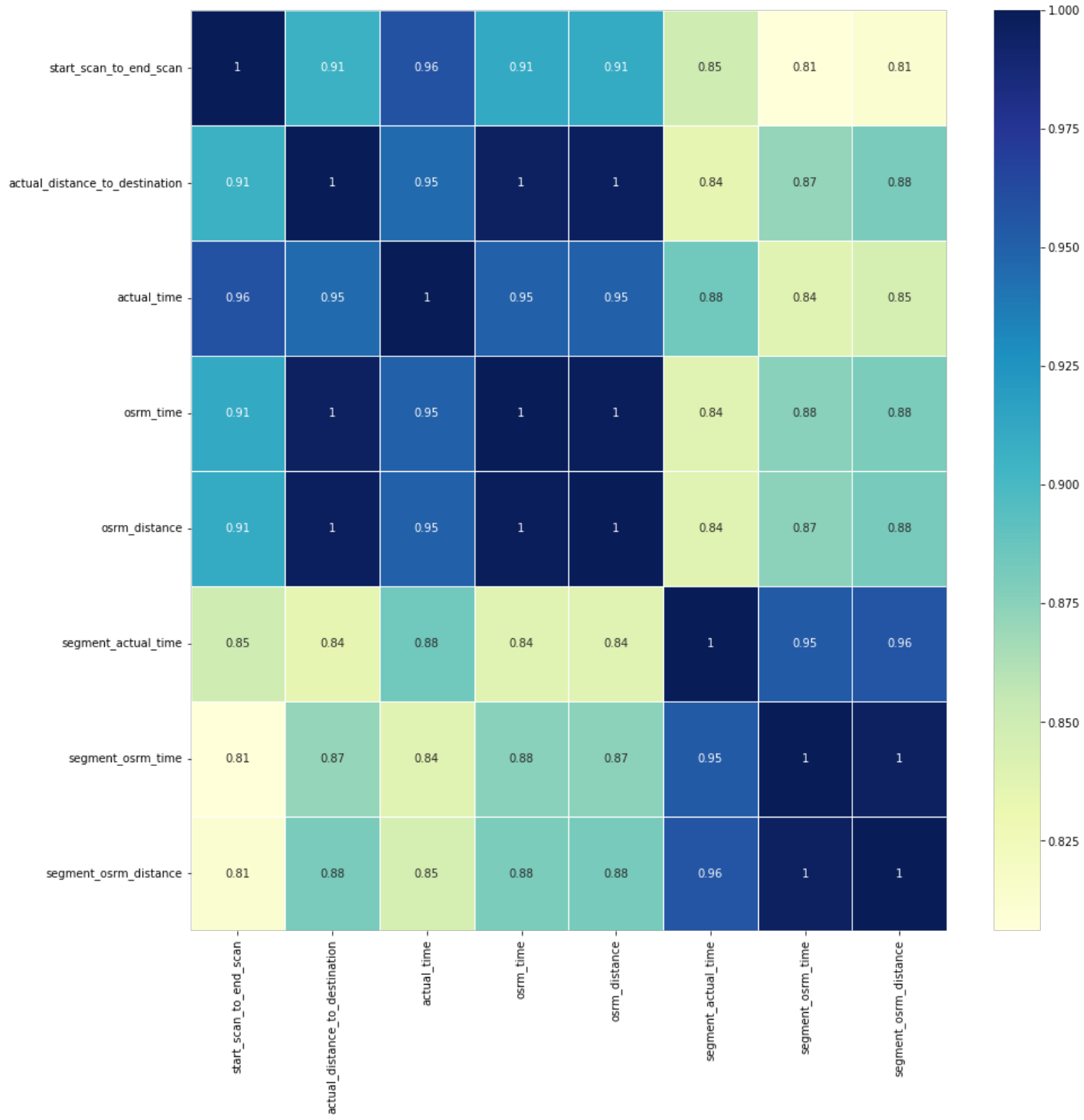




In [37]:

```
f, ax = plt.subplots(figsize =(15, 15))
sns.heatmap(df2[numerical].corr(), cmap ="YlGnBu", annot=True,linewidths = 0.5)
```

Out[37]:

<AxesSubplot:>




**all the numerical columns are slightly right skewed normal distribution and all the columns are highly positively correlated**

## Feature Creation

In [38]:

```
#converting to date time object
df2['trip_creation_time']= pd.to_datetime(df2['trip_creation_time'])
```

In [39]:

```
#creating year ,month and day columns
df2['trip_creation_year'] = df2['trip_creation_time'].dt.year
df2['trip_creation_month'] = df2['trip_creation_time'].dt.month_name(locale = 'English')
df2['trip_creation_day'] = df2['trip_creation_time'].dt.day_name(locale = 'English')
```

In [40]:

```
df2['trip_creation_year'].value_counts()
```

Out[40]:

```
2018    14787
Name: trip_creation_year, dtype: int64
```

In [41]:

```
df2['trip_creation_month'].value_counts(normalize=True)
```

Out[41]:

```
September    0.879895
October      0.120105
Name: trip_creation_month, dtype: float64
```

In [42]:

```
#function to get city name
def city(i):
    x=i.split('_')[0]
    x=x.split('(')[0]
    return x
```

In [43]:

```
# Extracting Source City name
df2['source_city']=df2['source_name'].apply(city)
```

In [44]:

```
## Extracting Destination City name
df2['destination_city']=df2['destination_name'].apply(city)
```

In [45]:

```
#function to get state name
def state(i):
    x=i.split('(')[1]
    return x[:-1]
```

In [46]:

```
df2.replace(to_replace ="Bangalore",
                 value ="Bengaluru",inplace=True)
```

In [47]:

```
# Extracting Source state name
df2['source_state']=df2['source_name'].apply(state)
```

In [48]:

```
# Extracting Destination state name
df2['destination_state']=df2['destination_name'].apply(state)
```

In [49]:

```
# Joining source city and destination city
df2['source_destination_city']=df2['source_city']+'_'+df2['destination_city']
```

In [50]:

```
# Joining source state and destination state
df2['source_destination_state']=df2['source_state']+'_'+df2['destination_state']
```

In [51]:

In [51]:

```
df2['source_destination_city'].value_counts().head(5)
```

Out[51]:

```
Bengaluru_Bengaluru      1376
Hyderabad_Hyderabad       398
Bhiwandi_Mumbai           332
Mumbai_Mumbai             264
Chandigarh_Chandigarh     248
Name: source_destination_city, dtype: int64
```

In [52]:

```
df2['source_destination_state'].value_counts().head(5)
```

Out[52]:

```
Maharashtra_Maharashtra    2406
Karnataka_Karnataka        2015
Tamil Nadu_Tamil Nadu      1016
Haryana_Haryana             867
Telangana_Telangana         655
Name: source_destination_state, dtype: int64
```

**Most of the trips are happening across same city and same state**

In [53]:

```
df2[['source_destination_state','route_type']].value_counts().head(5)
```

Out[53]:

```
source_destination_state  route_type
Maharashtra_Maharashtra   Carting       1978
Karnataka_Karnataka       Carting       1773
Tamil Nadu_Tamil Nadu     Carting        753
Haryana_Haryana           Carting        650
West Bengal_West Bengal   Carting        434
dtype: int64
```

In [54]:

```
df2[['source_destination_city','route_type']].value_counts().head(5)
```

Out[54]:

```
source_destination_city  route_type
Bengaluru_Bengaluru      Carting       1343
Bhiwandi_Mumbai          Carting        332
Hyderabad_Hyderabad      Carting        327
Mumbai_Mumbai            Carting        264
Gurgaon_Delhi            Carting        237
dtype: int64
```

**Most of the trips are happening across same city and same state and deliveries are happening through carting**

## Calculate the time taken between od_start_time and od_end_time and keep it as a feature. Drop the original columns, if required

In [56]:

```
df2['od_start_time']= pd.to_datetime(df2['od_start_time'])
df2['od_end_time']= pd.to_datetime(df2['od_end_time'])
```

In [57]:

```
df2['od_time_differnce_in_hrs']=((df2['od_end_time']-df2['od_start_time']).astype('timede
```

```
lta64[m]'))/60
```

## Longest trip

In [58]:

```
df2[df2['od_time_differnce_in_hrs']==max(df2['od_time_differnce_in_hrs'])]
```

Out[58]:

| | trip_uuid | data | trip_creation_time | route_type | source_center | source_name | destination_ce |
|---|---|---|---|---|---|---|---|
| 13577 | trip-153843695443252828 | test | 2018-10-01 23:35:54 | Carting | IND764071AAB | Pappadahandi_Central_DPP_2 (Orissa) | IND530012 |

1 rows × 28 columns

**The longest trip is recorded between papadahandi-Vishakapatnam cites.the trip took around 131 hrs which is 5.5 days**

## Shortest Trip

In [59]:

```
df2[df2['od_time_differnce_in_hrs']==min(df2['od_time_differnce_in_hrs'])]
```

Out[59]:

| | trip_uuid | data | trip_creation_time | route_type | source_center | source_name | destination_center | dest |
|---|---|---|---|---|---|---|---|---|
| 4575 | trip-153725231248161767 | training | 2018-09-18 06:31:52 | Carting | IND141010AAA | Ludhiana_DC (Punjab) | IND000000ACA | Ludhiana |

1 rows × 28 columns

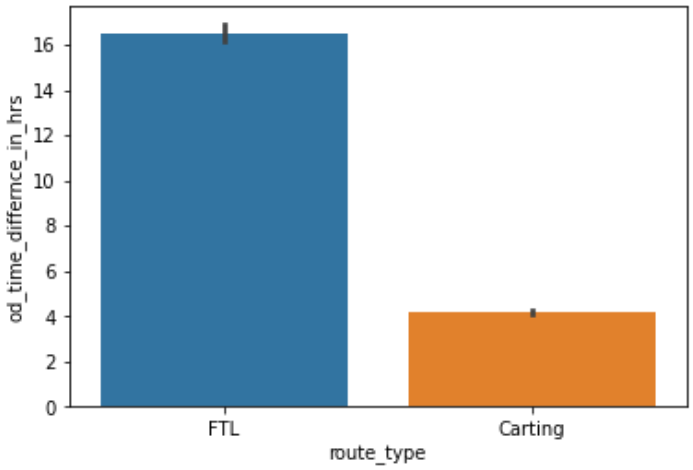
**shortest trip is from Ludhiana to Ludhiana in Punjab whick took 40 min to deliver**

In [60]:

```
sns.barplot(df2['route_type'],df2['od_time_differnce_in_hrs'])
```

Out[60]:

```
<AxesSubplot:xlabel='route_type', ylabel='od_time_differnce_in_hrs'>
```




**Deliveries are getting delivered fastly through carting than FTL**

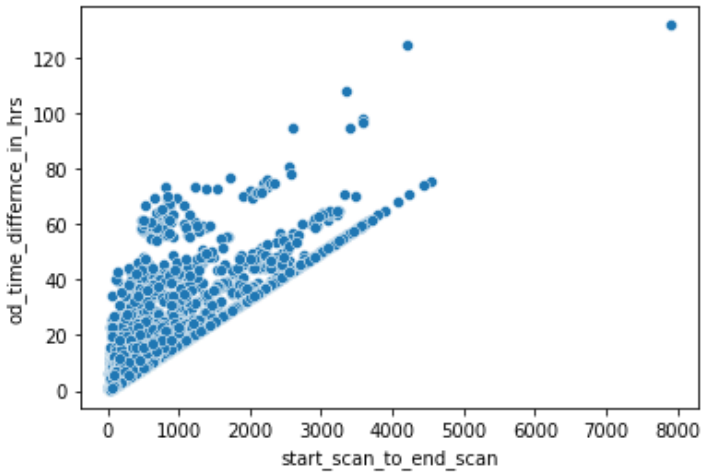
## Hypothesis testing/ visual analysis between start_scan_to_end_scan aggregated value and od_time_differnce_in_hrs

In [61]:

```
sns.scatterplot(df2['start_scan_to_end_scan'],df2['od_time_differnce_in_hrs'])
```

Out[61]:

```
<AxesSubplot:xlabel='start_scan_to_end_scan', ylabel='od_time_differnce_in_hrs'>
```




In [62]:

```
df2[["start_scan_to_end_scan", "od_time_differnce_in_hrs"]].corr()
```

Out[62]:

| | start_scan_to_end_scan | od_time_differnce_in_hrs |
|---|---|---|
| **start_scan_to_end_scan** | 1.000000 | 0.839586 |
| **od_time_differnce_in_hrs** | 0.839586 | 1.000000 |

In [63]:

```
Null_Hypothesis='start_scan_to_end_scan is equal to od_time_differnce_in_hrs'

Alternate_Hypothesis='Actual_start_scan_to_end_scantime is not equal to od_time_differnce
_in_hrs'

Level_of_significance=95
```

In [64]:

```
#two-sample T-Test
f,p=stats.ttest_ind(df2['start_scan_to_end_scan'],df2['od_time_differnce_in_hrs'],alterna
tive='two-sided')
print(f,p)
```

```
85.04805372177233 0.0
```

In [65]:

```
if p<0.05:
    print('We reject Null Hypothesis and go with '+Alternate_Hypothesis)
else:
    print('We accept Null Hypothesis '+Null_Hypothesis)
```

```
We reject Null Hypothesis and go with Actual_start_scan_to_end_scantime is not equal to o
d_time_differnce_in_hrs
```

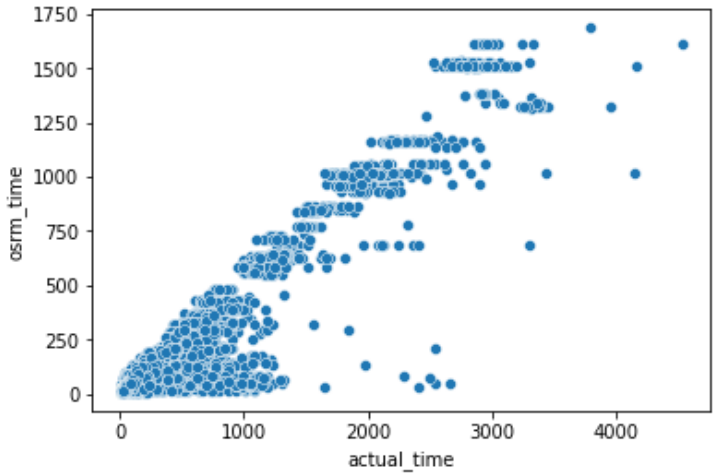
## Hypothesis testing/ visual analysis between actual_time aggregated value and OSRM time aggregated value

In [66]:

```
sns.scatterplot(x='actual_time', y='osrm_time', data=df2)
```

Out[66]:

```
<AxesSubplot:xlabel='actual_time', ylabel='osrm_time'>
```




In [67]:

```
df2[["actual_time", "osrm_time"]].corr()
```

Out[67]:

| | actual_time | osrm_time |
|---|---|---|
| **actual_time** | 1.000000 | 0.950013 |
| **osrm_time** | 0.950013 | 1.000000 |

In [68]:

```
Null_Hypothesis='Actual_time is equal to OSRM_time'

Alternate_Hypothesis='Actual_time is not equal to OSRM_time'

Level_of_significance=95
```

In [69]:

```
#two-sample T-Test
f,p=stats.ttest_ind(df2['actual_time'],df2['osrm_time'],alternative='two-sided')
print(f,p)
```

```
34.39952676204439 2.6617031367929406e-254
```

In [70]:

```
if p<0.05:
    print('We reject Null Hypothesis and go with '+Alternate_Hypothesis)
else:
    print('We accept Null Hypothesis '+Null_Hypothesis)
```

```
We reject Null Hypothesis and go with Actual_time is not equal to OSRM_time
```

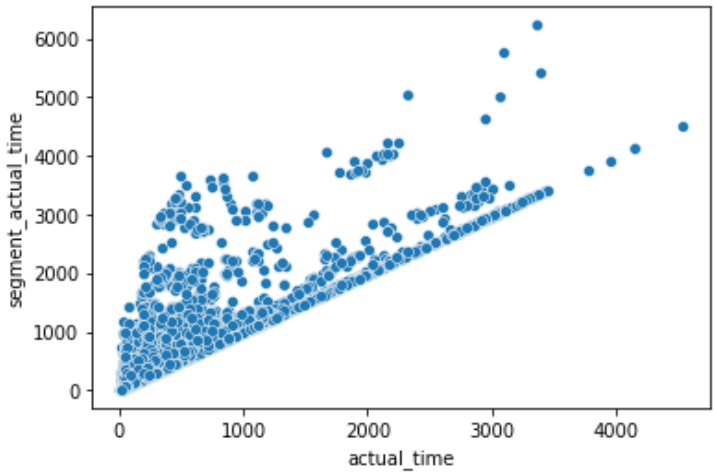
## Hypothesis testing/ visual analysis between actual_time aggregated value and segment actual time aggregated value

In [71]:

```
sns.scatterplot(x='actual_time', y='segment_actual_time', data=df2)
```

Out[71]:

<AxesSubplot:xlabel='actual_time', ylabel='segment_actual_time'>




In [72]:

```
df2[["actual_time", "segment_actual_time"]].corr()
```

Out[72]:

| | actual_time | segment_actual_time |
|---|---|---|
| **actual_time** | 1.000000 | 0.884709 |
| **segment_actual_time** | 0.884709 | 1.000000 |

In [73]:

```
Null_Hypothesis='Actual_time is equal to segment_actual_time'

Alternate_Hypothesis='Actual_time is not equal to segment_actual_time'

Level_of_significance=95
```

In [74]:

```
#two-sample T-Test
f,p=stats.ttest_ind(df2['actual_time'],df2['segment_actual_time'],alternative='two-sided'
)
print(f,p)
```

```
-17.241645469641675 2.729470100614497e-66
```

In [75]:

```
if p<0.05:
    print('We reject Null Hypothesis and go with '+Alternate_Hypothesis)
else:
    print('We accept Null Hypothesis '+Null_Hypothesis)
```

```
We reject Null Hypothesis and go with Actual_time is not equal to segment_actual_time
```

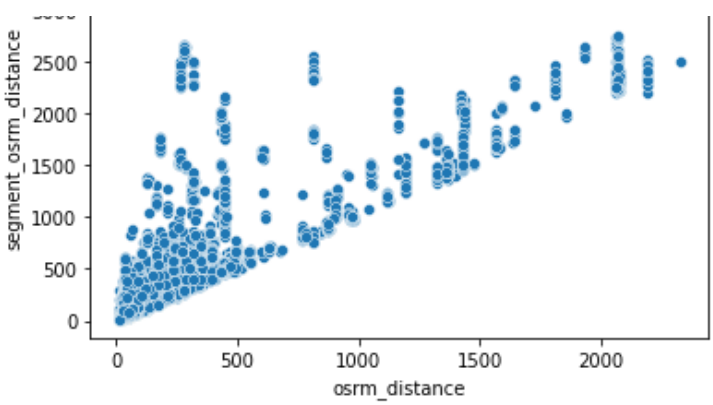
## Hypothesis testing/ visual analysis between osrm distance aggregated value and segment osrm distance aggregated value

In [76]:

```
sns.scatterplot(x='osrm_distance', y='segment_osrm_distance', data=df2)
```

Out[76]:

<AxesSubplot:xlabel='osrm_distance', ylabel='segment_osrm_distance'>




In [77]:

```
df2[["osrm_distance", "segment_osrm_distance"]].corr()
```

Out[77]:

| | osrm_distance | segment_osrm_distance |
|---|---|---|
| **osrm_distance** | 1.000000 | 0.881543 |
| **segment_osrm_distance** | 0.881543 | 1.000000 |

In [78]:

```
Null_Hypothesis='osrm_distance is equal to segment_osrm_distance'

Alternate_Hypothesis='osrm_distance is not equal to segment_osrm_distance'

Level_of_significance=95
```

In [79]:

```
#two-sample T-Test
f,p=stats.ttest_ind(df2['osrm_distance'],df2['segment_osrm_distance'],alternative='two-si
ded')
print(f,p)
```

```
-19.981776650079656 3.0376487186424705e-88
```

In [80]:

```
if p<0.05:
    print('We reject Null Hypothesis and go with '+Alternate_Hypothesis)
else:
    print('We accept Null Hypothesis '+Null_Hypothesis)
```

```
We reject Null Hypothesis and go with osrm_distance is not equal to segment_osrm_distance
```

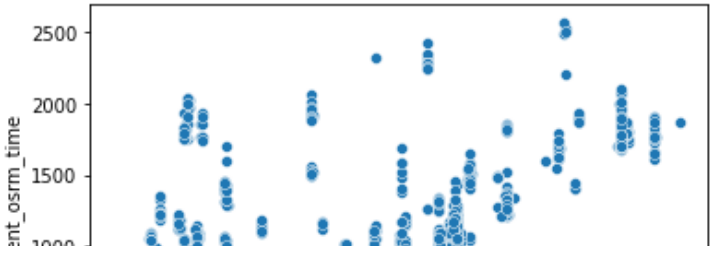
## Hypothesis testing/ visual analysis between osrm time aggregated value and segment osrm time aggregated value

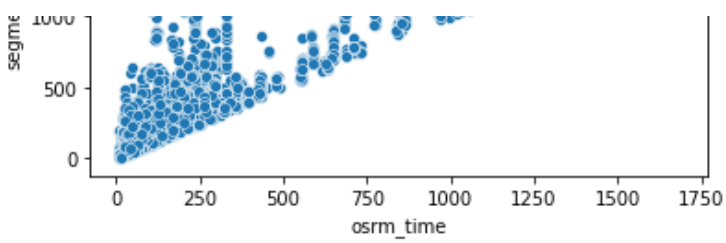
In [81]:

```
sns.scatterplot(x='osrm_time', y='segment_osrm_time', data=df2)
```

Out[81]:

```
<AxesSubplot:xlabel='osrm_time', ylabel='segment_osrm_time'>
```






In [82]:

```
df2[["osrm_time", "segment_osrm_time"]].corr()
```

Out[82]:

| | osrm_time | segment_osrm_time |
|---|---|---|
| **osrm_time** | 1.000000 | 0.875547 |
| **segment_osrm_time** | 0.875547 | 1.000000 |

In [83]:

```
Null_Hypothesis='osrm_time is equal to segment_osrm_time'

Alternate_Hypothesis='osrm_time is not equal to segment_osrm_time'

Level_of_significance=95
```

In [84]:

```
#2 sample T Test
f,p=stats.ttest_ind(df2['osrm_time'],df2['segment_osrm_time'],alternative='two-sided')
print(f,p)
```

```
-22.92172635893408 2.8576015291925124e-115
```

In [85]:

```
if p<0.05:
    print('We reject Null Hypothesis and go with '+Alternate_Hypothesis)
else:
    print('We accept Null Hypothesis '+Null_Hypothesis)
```

```
We reject Null Hypothesis and go with osrm_time is not equal to segment_osrm_time
```

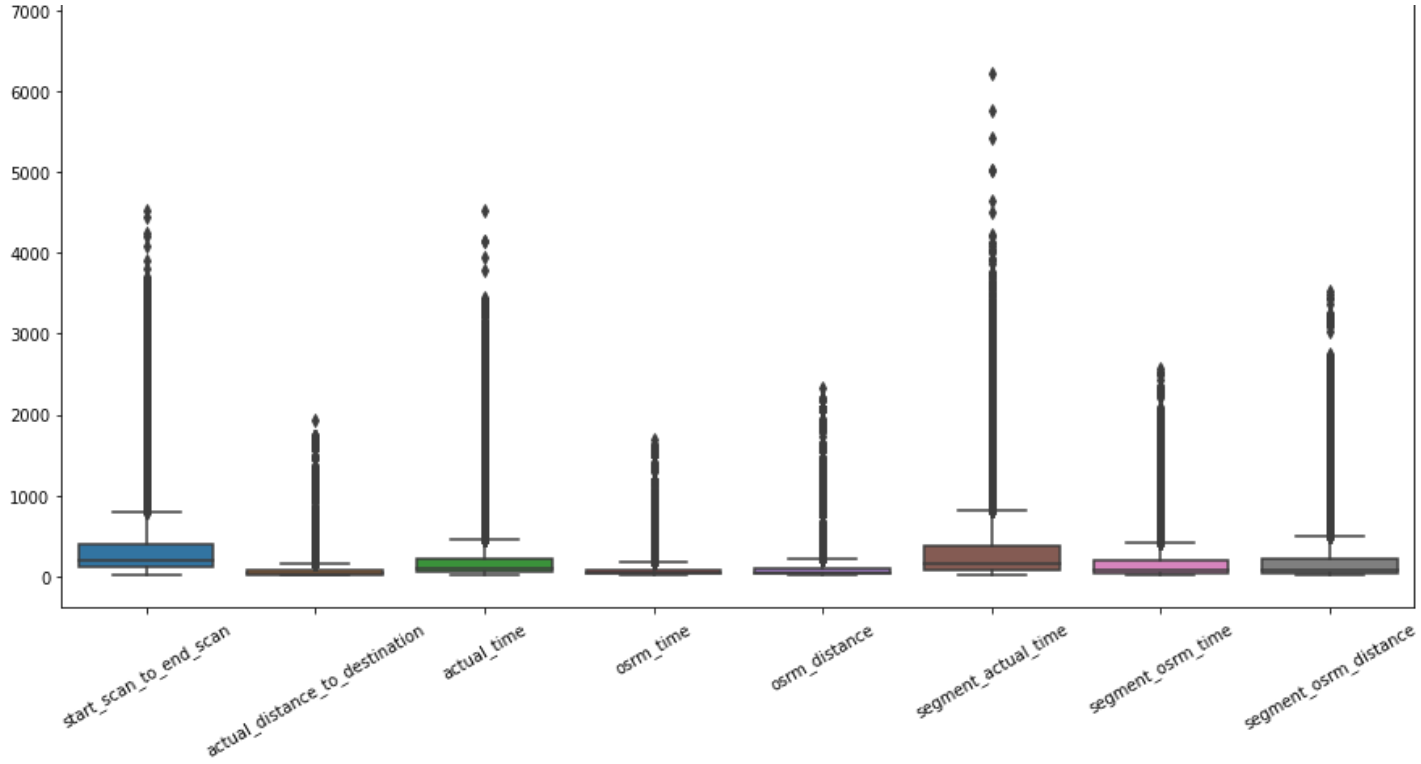
## Outlier Detection

In [86]:

```
f, ax = plt.subplots(figsize =(15,8))
sns.boxplot(data=df2[['start_scan_to_end_scan','actual_distance_to_destination','actual_time','osrm_time','osrm_distance','segment_actual_time','segment_osrm_time','segment_osrm_distance']])
plt.xticks(rotation=30)
```

Out[86]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7]),
 [Text(0, 0, 'start_scan_to_end_scan'),
  Text(1, 0, 'actual_distance_to_destination'),
  Text(2, 0, 'actual_time'),
  Text(3, 0, 'osrm_time'),
  Text(4, 0, 'osrm_distance'),
  Text(5, 0, 'segment_actual_time'),
  Text(6, 0, 'segment_osrm_time'),
  Text(7, 0, 'segment_osrm_distance')])
```




**There are lot of outliers in given data**

In [87]:

```
cols=['start_scan_to_end_scan','actual_distance_to_destination','actual_time','osrm_time'
,'osrm_distance','segment_actual_time','segment_osrm_time','segment_osrm_distance']
```

## Removing outliers

In [88]:

```
#Removing outliers
for i in cols:
    q3=df2[i].quantile(0.75)
    q1=df2[i].quantile(0.25)
    iqr=q3-q1
    df2=df2[(df2[i]>q1-(1.5*iqr)) & (df2[i]<q3+(1.5*iqr))]
```

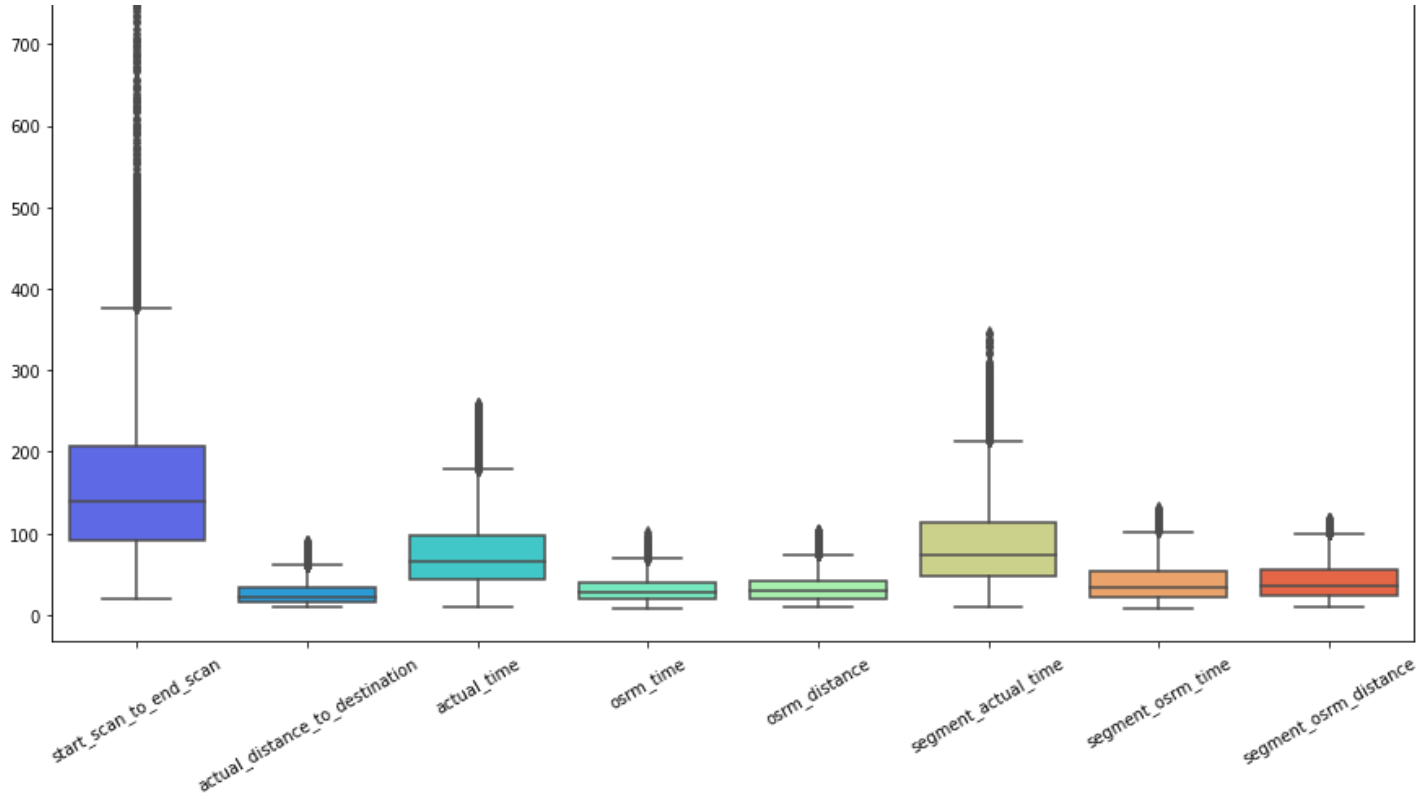
## Visualization after removing outliers

In [89]:

```
f, ax = plt.subplots(figsize =(15,8))
sns.boxplot(data=df2[['start_scan_to_end_scan','actual_distance_to_destination','actual_t
ime','osrm_time','osrm_distance','segment_actual_time','segment_osrm_time','segment_osrm_
distance']],palette='rainbow')
plt.xticks(rotation=30)
```

Out[89]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7]),
 [Text(0, 0, 'start_scan_to_end_scan'),
  Text(1, 0, 'actual_distance_to_destination'),
  Text(2, 0, 'actual_time'),
  Text(3, 0, 'osrm_time'),
  Text(4, 0, 'osrm_distance'),
  Text(5, 0, 'segment_actual_time'),
  Text(6, 0, 'segment_osrm_time'),
  Text(7, 0, 'segment_osrm_distance')])
```




## one-hot encoding of categorical variables

In [90]:

```
columns=list(df2.columns)
```

In [91]:

```
#seperating categorical variables
columns=list(df2.columns)
categorical=[]
for  i in columns:
    if df2[i].dtype=='object':
        categorical.append(i)
```

In [92]:

```
print(categorical)
```

```
['trip_uuid', 'data', 'route_type', 'source_center', 'source_name', 'destination_center',
'destination_name', 'trip_creation_month', 'trip_creation_day', 'source_city', 'destinati
on_city', 'source_state', 'destination_state', 'source_destination_city', 'source_destina
tion_state']
```

In [93]:

```
# creating instance of one-hot-encoder
enc = OneHotEncoder(handle_unknown='ignore')
```

### One Hot Encoding for Categorical Variables

In [94]:

```
#One Hot Encoding for all Categorical Variables
for i in categorical:
    print(i+' one-hot-encoding')
    enc_df = pd.DataFrame(enc.fit_transform(df2[[i]]).toarray())
    print(enc_df)
```

```
trip_uuid one-hot-encoding
      0     1     2     3     4     5     6     7     8     9     ...   8452  \
0      1.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0          0.0
```

```
0      1.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
1      0.0   1.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
2      0.0   0.0   1.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
3      0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
4      0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
...    ...   ...   ...   ...   ...   ...   ...   ...   ...   ...  ...   ...
8457   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
8458   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
8459   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
8460   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0
8461   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  ...   0.0

      8453  8454  8455  8456  8457  8458  8459  8460  8461
0      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
1      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
2      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
3      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4      0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
...    ...   ...   ...   ...   ...   ...   ...   ...   ...
8457   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0
8458   0.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0
8459   0.0   0.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0
8460   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0   0.0
8461   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0

[8462 rows x 8462 columns]
data one-hot-encoding
        0    1
0     0.0  1.0
1     0.0  1.0
2     0.0  1.0
3     0.0  1.0
4     0.0  1.0
...   ...  ...
8457  1.0  0.0
8458  1.0  0.0
8459  1.0  0.0
8460  1.0  0.0
8461  1.0  0.0

[8462 rows x 2 columns]
route_type one-hot-encoding
        0    1
0     1.0  0.0
1     1.0  0.0
2     1.0  0.0
3     1.0  0.0
4     1.0  0.0
...   ...  ...
8457  1.0  0.0
8458  1.0  0.0
8459  1.0  0.0
8460  1.0  0.0
8461  0.0  1.0

[8462 rows x 2 columns]
source_center one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  599  600  601  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      602  603  604  605  606  607  608
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 609 columns]
source_name one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  599  600  601  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      602  603  604  605  606  607  608
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 609 columns]
destination_center one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  681  682  683  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      684  685  686  687  688  689  690
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 691 columns]
destination_name one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  681  682  683  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
```

```
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      684  685  686  687  688  689  690
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 691 columns]
trip_creation_month one-hot-encoding
        0    1
0     0.0  1.0
1     0.0  1.0
2     0.0  1.0
3     0.0  1.0
4     0.0  1.0
...   ...  ...
8457  1.0  0.0
8458  1.0  0.0
8459  1.0  0.0
8460  1.0  0.0
8461  1.0  0.0

[8462 rows x 2 columns]
trip_creation_day one-hot-encoding
        0    1    2    3    4    5    6
0     0.0  0.0  0.0  0.0  0.0  0.0  1.0
1     0.0  0.0  0.0  0.0  0.0  0.0  1.0
2     0.0  0.0  0.0  0.0  0.0  0.0  1.0
3     0.0  0.0  0.0  0.0  0.0  0.0  1.0
4     0.0  0.0  0.0  0.0  0.0  0.0  1.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  1.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  1.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  1.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  1.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  1.0

[8462 rows x 7 columns]
source_city one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  434  435  436  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      437  438  439  440  441  442  443
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 444 columns]
destination_city one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  527  528  529  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      530  531  532  533  534  535  536
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 537 columns]
source_state one-hot-encoding
       0    1    2    3    4    5    6    7    8    9    ...   16   17   18  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

       19   20   21   22   23   24   25
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  1.0  0.0  0.0  0.0  0.0
3     0.0  0.0  1.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  1.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  1.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 26 columns]
destination_state one-hot-encoding
       0    1    2    3    4    5    6    7    8    9    ...   18   19   20  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
```

```
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

       21   22   23   24   25   26   27
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  1.0  0.0  0.0  0.0  0.0
3     0.0  0.0  1.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  1.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  1.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 28 columns]
source_destination_city one-hot-encoding
      0    1    2    3    4    5    6    7    8    9    ...  786  787  788  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

      789  790  791  792  793  794  795
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 796 columns]
source_destination_state one-hot-encoding
       0    1    2    3    4    5    6    7    8    9    ...   40   41   42  \
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  1.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  1.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

       43   44   45   46   47   48   49
0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
1     0.0  0.0  0.0  0.0  0.0  0.0  0.0
2     0.0  0.0  0.0  0.0  0.0  0.0  0.0
3     0.0  0.0  0.0  0.0  0.0  0.0  0.0
4     0.0  0.0  0.0  0.0  0.0  0.0  0.0
...   ...  ...  ...  ...  ...  ...  ...
8457  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8458  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
8459  0.0  0.0  0.0  0.0  0.0  0.0  0.0
8460  0.0  0.0  0.0  0.0  1.0  0.0  0.0
8461  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[8462 rows x 50 columns]
```

In [95]:

```
#One Hot Encoding for data and route_type variables
cat=['data' ,'route_type']
for i in cat:
    print(i+' one-hot-encoding')
    one_hot_encoded_data = pd.get_dummies(df2, columns = cat)
    print(one_hot_encoded_data)
```

```
data one-hot-encoding
                    trip_uuid  trip_creation_time source_center  \
1      trip-153671042288605164 2018-09-12 00:00:23  IND572101AAA
3      trip-153671046011330457 2018-09-12 00:01:00  IND400072AAB
5      trip-153671055416136166 2018-09-12 00:02:34  IND600116AAB
6      trip-153671066201138152 2018-09-12 00:04:22  IND600044AAD
7      trip-153671066826362165 2018-09-12 00:04:28  IND560043AAC
...                        ...                  ...           ...
14781  trip-153861091843037040 2018-10-03 23:55:18  IND400072AAB
14782  trip-153861095625827784 2018-10-03 23:55:56  IND160002AAC
14783  trip-153861104386292051 2018-10-03 23:57:24  IND121004AAB
14784  trip-153861106442901555 2018-10-03 23:57:44  IND209304AAA
14786  trip-153861118270144424 2018-10-03 23:59:43  IND583201AAA

                            source_name destination_center  \
1            Tumkur_Veersagr_I (Karnataka)       IND562101AAA
3                 Mumbai Hub (Maharashtra)       IND401104AAA
5            Chennai_Porur_DPC (Tamil Nadu)       IND602105AAB
6         Chennai_Chrompet_DPC (Tamil Nadu)       IND600048AAA
7                 HBR Layout PC (Karnataka)       IND560043AAC
...                                     ...                ...
14781             Mumbai Hub (Maharashtra)       IND401104AAA
14782        Chandigarh_Mehmdpur_H (Punjab)       IND160002AAC
14783          FBD_Balabhgarh_DPC (Haryana)       IND121004AAA
14784  Kanpur_Central_H_6 (Uttar Pradesh)       IND209304AAA
14786                   Hospet (Karnataka)       IND583101AAA

                          destination_name       od_start_time  \
1          Chikblapur_ShntiSgr_D (Karnataka) 2018-09-12 00:00:23
3             Mumbai_MiraRd_IP (Maharashtra) 2018-09-12 00:01:00
5      Chennai_Sriperumbudur_Dc (Tamil Nadu) 2018-09-12 00:02:34
6           Chennai_Vandalur_Dc (Tamil Nadu) 2018-09-12 00:04:22
7                  HBR Layout PC (Karnataka) 2018-09-12 00:04:28
...                                      ...                 ...
14781         Mumbai_MiraRd_IP (Maharashtra) 2018-10-03 23:55:18
14782         Chandigarh_Mehmdpur_H (Punjab) 2018-10-03 23:55:56
14783         Faridabad_Blbgarh_DC (Haryana) 2018-10-03 23:57:24
14784     Kanpur_Central_H_6 (Uttar Pradesh) 2018-10-03 23:57:44
14786                 Bellary_Dc (Karnataka) 2018-10-04 02:51:45

             od_end_time  start_scan_to_end_scan  \
1      2018-09-12 03:02:00                      58
3      2018-09-12 01:41:30                     100
5      2018-09-12 03:13:03                      60
6      2018-09-12 01:42:22                      98
7      2018-09-12 03:00:55                      78
...                    ...                     ...
14781  2018-10-04 01:23:31                      88
14782  2018-10-04 06:41:25                     105
14783  2018-10-04 00:57:59                      60
14784  2018-10-04 06:59:52                     248
14786  2018-10-04 08:46:09                     287

       actual_distance_to_destination  ...  destination_city   source_state  \
1                           24.644021  ...        Chikblapur      Karnataka
3                           17.175274  ...            Mumbai    Maharashtra
5                           15.325529  ...           Chennai     Tamil Nadu
```

```
5                        15.523529  ...           Chennai     Tamil Nadu
6                         9.100510  ...           Chennai     Tamil Nadu
7                        12.352948  ...    HBR Layout PC       Karnataka
...                            ...  ...               ...             ...
14781                    17.760248  ...            Mumbai    Maharashtra
14782                    31.261599  ...         Chandigarh         Punjab
14783                    15.513784  ...          Faridabad        Haryana
14784                    19.349008  ...             Kanpur  Uttar Pradesh
14786                    40.546740  ...            Bellary      Karnataka

      destination_state        source_destination_city  \
1             Karnataka             Tumkur_Chikblapur
3           Maharashtra            Mumbai Hub _Mumbai
5             Tamil Nadu               Chennai_Chennai
6             Tamil Nadu               Chennai_Chennai
7              Karnataka  HBR Layout PC _HBR Layout PC
...                  ...                          ...
14781       Maharashtra            Mumbai Hub _Mumbai
14782            Punjab         Chandigarh_Chandigarh
14783           Haryana                 FBD_Faridabad
14784     Uttar Pradesh                 Kanpur_Kanpur
14786         Karnataka               Hospet _Bellary

         source_destination_state  od_time_differnce_in_hrs  data_test  \
1              Karnataka_Karnataka                  3.016667          0
3           Maharashtra_Maharashtra                 1.666667          0
5             Tamil Nadu_Tamil Nadu                 3.166667          0
6             Tamil Nadu_Tamil Nadu                 1.633333          0
7               Karnataka_Karnataka                 2.933333          0
...                             ...                      ...        ...
14781       Maharashtra_Maharashtra                 1.466667          1
14782                 Punjab_Punjab                 6.750000          1
14783               Haryana_Haryana                 1.000000          1
14784  Uttar Pradesh_Uttar Pradesh                  7.033333          1
14786           Karnataka_Karnataka                 5.900000          1

      data_training route_type_Carting route_type_FTL
1                 1                  1              0
3                 1                  1              0
5                 1                  1              0
6                 1                  1              0
7                 1                  1              0
...             ...                ...            ...
14781             0                  1              0
14782             0                  1              0
14783             0                  1              0
14784             0                  1              0
14786             0                  0              1

[8462 rows x 30 columns]
route_type one-hot-encoding
                    trip_uuid  trip_creation_time source_center  \
1      trip-153671042288605164 2018-09-12 00:00:23  IND572101AAA
3      trip-153671046011330457 2018-09-12 00:01:00  IND400072AAB
5      trip-153671055416136166 2018-09-12 00:02:34  IND600116AAB
6      trip-153671066201138152 2018-09-12 00:04:22  IND600044AAD
7      trip-153671066826362165 2018-09-12 00:04:28  IND560043AAC
...                        ...                  ...           ...
14781  trip-153861091843037040 2018-10-03 23:55:18  IND400072AAB
14782  trip-153861095625827784 2018-10-03 23:55:56  IND160002AAC
14783  trip-153861104386292051 2018-10-03 23:57:24  IND121004AAB
14784  trip-153861106442901555 2018-10-03 23:57:44  IND209304AAA
14786  trip-153861118270144424 2018-10-03 23:59:43  IND583201AAA

                           source_name destination_center  \
1           Tumkur_Veersagr_I (Karnataka)       IND562101AAA
3                Mumbai Hub (Maharashtra)       IND401104AAA
5           Chennai_Porur_DPC (Tamil Nadu)       IND602105AAB
6         Chennai_Chrompet_DPC (Tamil Nadu)      IND600048AAA
7                 HBR Layout PC (Karnataka)      IND560043AAC
...                                     ...               ...
14781            Mumbai Hub (Maharashtra)       IND401104AAA
14782      Chandigarh_Mehmdpur_H (Punjab)       IND160002AAC
```

```
14782      Chandigarh_Mehmdpur_H (Punjab)       IND160002AAC
14783        FBD_Balabhgarh_DPC (Haryana)       IND121004AAA
14784  Kanpur_Central_H_6 (Uttar Pradesh)       IND209304AAA
14786                 Hospet (Karnataka)       IND583101AAA

                          destination_name       od_start_time  \
1          Chikblapur_ShntiSgr_D (Karnataka) 2018-09-12 00:00:23
3             Mumbai_MiraRd_IP (Maharashtra) 2018-09-12 00:01:00
5      Chennai_Sriperumbudur_Dc (Tamil Nadu) 2018-09-12 00:02:34
6           Chennai_Vandalur_Dc (Tamil Nadu) 2018-09-12 00:04:22
7                 HBR Layout PC (Karnataka) 2018-09-12 00:04:28
...                                     ...                 ...
14781         Mumbai_MiraRd_IP (Maharashtra) 2018-10-03 23:55:18
14782         Chandigarh_Mehmdpur_H (Punjab) 2018-10-03 23:55:56
14783         Faridabad_Blbgarh_DC (Haryana) 2018-10-03 23:57:24
14784     Kanpur_Central_H_6 (Uttar Pradesh) 2018-10-03 23:57:44
14786                Bellary_Dc (Karnataka) 2018-10-04 02:51:45

              od_end_time  start_scan_to_end_scan  \
1     2018-09-12 03:02:00                      58
3     2018-09-12 01:41:30                     100
5     2018-09-12 03:13:03                      60
6     2018-09-12 01:42:22                      98
7     2018-09-12 03:00:55                      78
...                   ...                     ...
14781 2018-10-04 01:23:31                      88
14782 2018-10-04 06:41:25                     105
14783 2018-10-04 00:57:59                      60
14784 2018-10-04 06:59:52                     248
14786 2018-10-04 08:46:09                     287

       actual_distance_to_destination  ...  destination_city   source_state  \
1                           24.644021  ...        Chikblapur      Karnataka
3                           17.175274  ...            Mumbai    Maharashtra
5                           15.325529  ...           Chennai     Tamil Nadu
6                            9.100510  ...           Chennai     Tamil Nadu
7                           12.352948  ...    HBR Layout PC       Karnataka
...                               ...  ...               ...             ...
14781                       17.760248  ...            Mumbai    Maharashtra
14782                       31.261599  ...         Chandigarh         Punjab
14783                       15.513784  ...          Faridabad        Haryana
14784                       19.349008  ...             Kanpur  Uttar Pradesh
14786                       40.546740  ...            Bellary      Karnataka

       destination_state        source_destination_city  \
1              Karnataka              Tumkur_Chikblapur
3            Maharashtra             Mumbai Hub _Mumbai
5             Tamil Nadu                Chennai_Chennai
6             Tamil Nadu                Chennai_Chennai
7              Karnataka  HBR Layout PC _HBR Layout PC
...                  ...                           ...
14781        Maharashtra             Mumbai Hub _Mumbai
14782             Punjab          Chandigarh_Chandigarh
14783            Haryana                  FBD_Faridabad
14784      Uttar Pradesh                  Kanpur_Kanpur
14786          Karnataka                Hospet _Bellary

          source_destination_state  od_time_differnce_in_hrs  data_test  \
1             Karnataka_Karnataka                  3.016667          0
3          Maharashtra_Maharashtra                 1.666667          0
5            Tamil Nadu_Tamil Nadu                 3.166667          0
6            Tamil Nadu_Tamil Nadu                 1.633333          0
7              Karnataka_Karnataka                 2.933333          0
...                            ...                      ...        ...
14781      Maharashtra_Maharashtra                 1.466667          1
14782                Punjab_Punjab                 6.750000          1
14783              Haryana_Haryana                 1.000000          1
14784  Uttar Pradesh_Uttar Pradesh                 7.033333          1
14786          Karnataka_Karnataka                 5.900000          1

      data_training route_type_Carting route_type_FTL
1                 1                  1              0
3                 1                  1              0
```

```
5                 1                 1                 0
6                 1                 1                 0
7                 1                 1                 0
...             ...               ...               ...
14781             0                 1                 0
14782             0                 1                 0
14783             0                 1                 0
14784             0                 1                 0
14786             0                 0                 1

[8462 rows x 30 columns]
```

## Normalize/ Standardize the numerical columns using MinMaxScaler

In [96]:

```
numerical=['start_scan_to_end_scan',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'segment_actual_time',
 'segment_osrm_time',
 'segment_osrm_distance','od_time_differnce_in_hrs']
```

In [97]:

```
for i in numerical:
    df2[i] = MinMaxScaler().fit_transform(df2[[i]])
```

In [98]:

```
df2
```

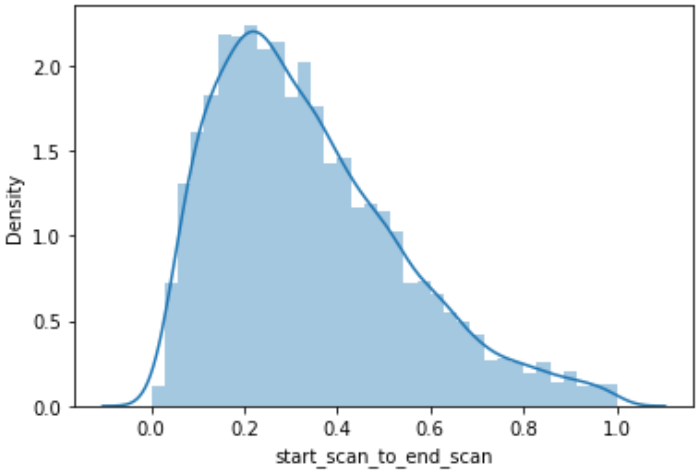
Out[98]:

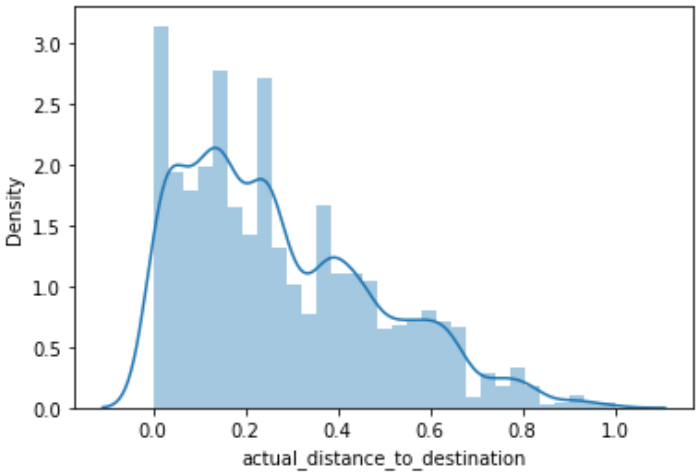
| | trip_uuid | data | trip_creation_time | route_type | source_center | source_name | destination_cer |
|---|---|---|---|---|---|---|---|
| 1 | trip-153671042288605164 | training | 2018-09-12 00:00:23 | Carting | IND572101AAA | Tumkur_Veersagr_I (Karnataka) | IND562101A |
| 3 | trip-153671046011330457 | training | 2018-09-12 00:01:00 | Carting | IND400072AAB | Mumbai Hub (Maharashtra) | IND401104A |
| 5 | trip-153671055416136166 | training | 2018-09-12 00:02:34 | Carting | IND600116AAB | Chennai_Porur_DPC (Tamil Nadu) | IND602105A |
| 6 | trip-153671066201138152 | training | 2018-09-12 00:04:22 | Carting | IND600044AAD | Chennai_Chrompet_DPC (Tamil Nadu) | IND600048A |
| 7 | trip-153671066826362165 | training | 2018-09-12 00:04:28 | Carting | IND560043AAC | HBR Layout PC (Karnataka) | IND560043A |
| ... | ... | ... | ... | ... | ... | ... | |
| 14781 | trip-153861091843037040 | test | 2018-10-03 23:55:18 | Carting | IND400072AAB | Mumbai Hub (Maharashtra) | IND401104A |
| 14782 | trip-153861095625827784 | test | 2018-10-03 23:55:56 | Carting | IND160002AAC | Chandigarh_Mehmdpur_H (Punjab) | IND160002A |
| 14783 | trip-153861104386292051 | test | 2018-10-03 23:57:24 | Carting | IND121004AAB | FBD_Balabhgarh_DPC (Haryana) | IND121004A |
| 14784 | trip-153861106442901555 | test | 2018-10-03 23:57:44 | Carting | IND209304AAA | Kanpur_Central_H_6 (Uttar Pradesh) | IND209304A |
| 14786 | trip-153861118270144424 | test | 2018-10-03 23:59:43 | FTL | IND583201AAA | Hospet (Karnataka) | IND583101A |

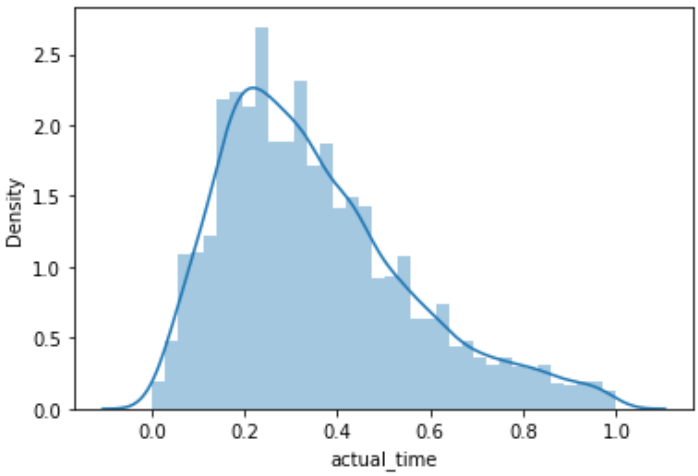
8462 rows × 28 columns

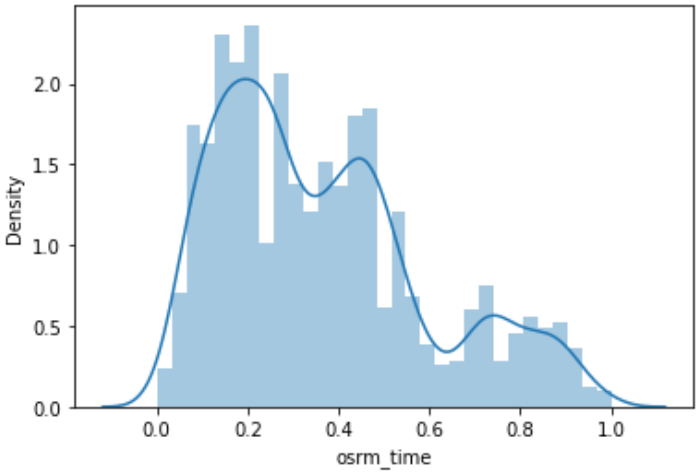
In [397]:

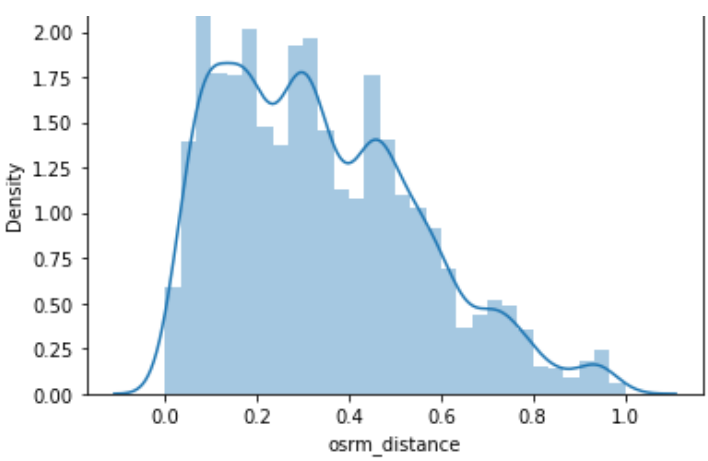
```
for i in numerical:
    plt.figure()              # <==================== here!
    sns.distplot(df2[i])
```

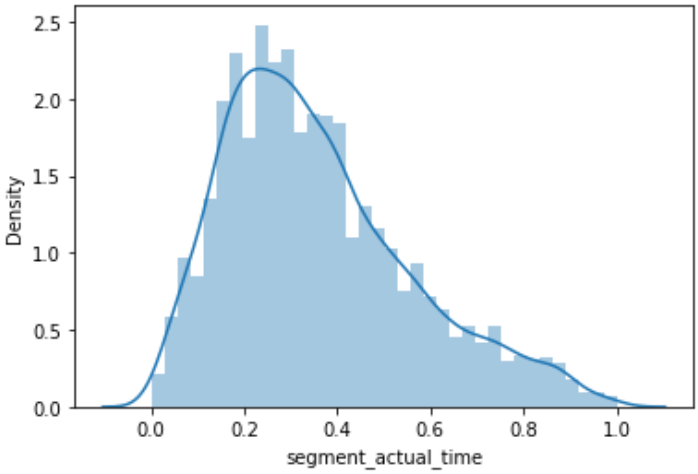


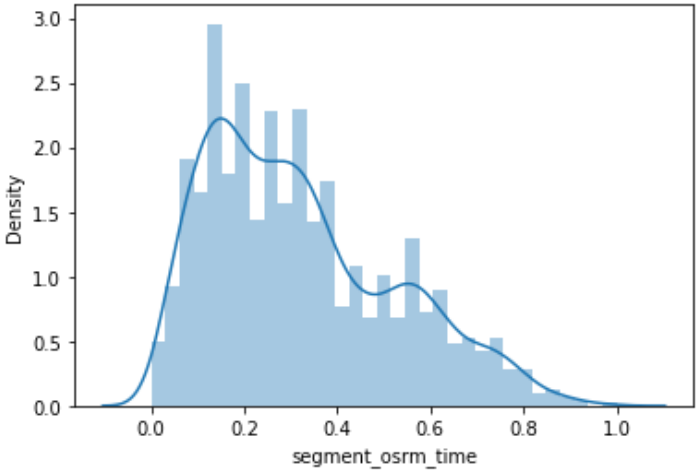



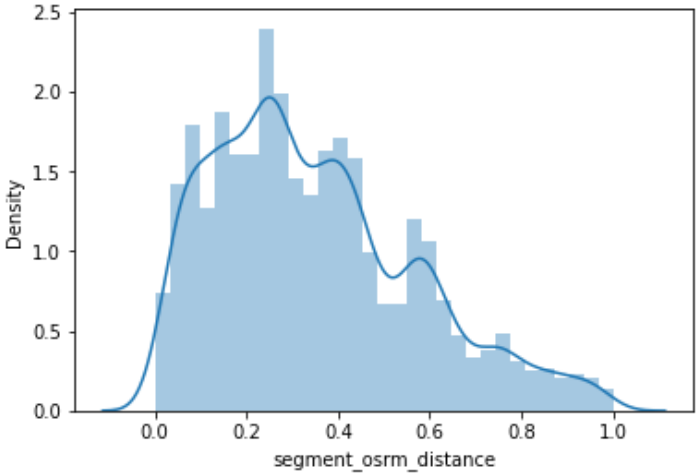



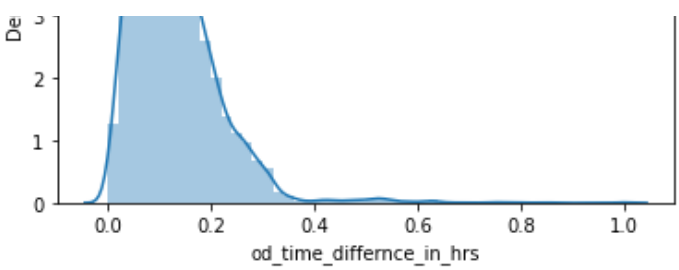




Density
osrm_distance


Density
segment_actual_time


Density
segment_osrm_time


Density
segment_osrm_distance




In [ ]: