Problem statement:

Delhivery is the largest and fastest-growing fully integrated player in India by revenue in Fiscal 2021. They aim to build the operating system for commerce, through a combination of world-class infrastructure, logistics operations of the highest quality, and cutting-edge engineering and technology capabilities.Now i have to help their datascience team to give insights and recommendations by Clean, sanitize and manipulate data to get useful features out of raw fields of the given dataset.I have to make sense out of the raw data and help the data science team to build forecasting models on it

Business Insights

```
* In most of the cases the predicted time is less compared to actual time taken to
deliver the product
* Median time taken after scanning the product at start and scanning at end is alm
ost 449 mins
* In most of the cases the predicted distance is more than the actual distance
* Most of the products delivered under 132 mins
* Most of the products delivered in metropolitan cities like bengaluru,etc
* Less no of products were delivered in places like union territories
```

Recommendations

```
* Comapny should optimize their engine to give more accurate predicted distances a
nd times.
* Company should focus on reaching to ground level in every place.
* Company should provide some offers for areas where there are less no of products
delivered.
* Company sholud provide more faster delivery as much as possible.
* Company should establish multiple dark stores in every area as possible in metro
politan cities.
* company should advertise more in areas where there is more rate of customers com
ing.
```

In [ ]:

```

```

In [140]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
from scipy import stats
import statsmodels.api as sm
from statsmodels.stats.weightstats import ztest
```

In [2]:

```
data=pd.read_csv('delhivery_data.csv')
```

In [3]:

```
data.head()
```

Out[3]:

amp	actual_distance_to_destination	actual_time	osrm_time	osrm_distance	factor	segment_ac
9-20 7:55	10.435660	14.0	11.0	11.9653	1.272727	
9-20 7:55	18.936842	24.0	20.0	21.7243	1.200000	
9-20 586	27.637279	40.0	28.0	32.5395	1.428571	
9-20 9:57	36.118028	62.0	40.0	45.5620	1.550000	
9-20 3:55	39.386040	68.0	44.0	54.2181	1.545455	

In [ ]:

```

```

In [4]:

```
data.shape
```

Out[4]:

```
(144867, 24)
```

In [5]:

```
data.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 [6]:

```
data.dtypes
```

Out[6]:

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

In [7]:

```
data.describe()
```

Out[7]:

actual_time	osrm_time	osrm_distance	factor	segment_actual_time	segment_osri
144867.000000	144867.000000	144867.000000	144867.000000	144867.000000	144867.
416.927527	213.868272	284.771297	2.120107	36.196111	18.
598.103621	308.011085	421.119294	1.715421	53.571158	14.
9.000000	6.000000	9.008200	0.144000	-244.000000	0.
51.000000	27.000000	29.914700	1.604264	20.000000	11.
132.000000	64.000000	78.525800	1.857143	29.000000	17.
513.000000	257.000000	343.193250	2.213483	40.000000	22.
4532.000000	1686.000000	2326.199100	77.387097	3051.000000	1611.

In [8]:

```
data.describe(include='all')
```

Out[8]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	sc
count	144867	144867	144867	144867	144867	
unique	2	14817	1504	2	14817	
top	training	2018-09-19 04:07:34.091798	thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069f...	FTL	trip-153715938946690081	IN
freq	104858	101	1812	99660	101	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

11 rows × 24 columns

Statistical Summary

```
*Here median start_scan_to_end_scan    time is less than 449 but there are some or
ders which took 7898 mins.
* Median actual time taken is under 52 but the median estimated time is 27.
```

In [ ]:

In [9]:

```
data.isna().sum()
```

```
source_center                    0
source_name                    293
destination_center               0
destination_name               261
od_start_time                    0
od_end_time                      0
start_scan_to_end_scan           0
is_cutoff                        0
cutoff_factor                    0
cutoff_timestamp                 0
actual_distance_to_destination   0
actual_time                      0
osrm_time                        0
osrm_distance                    0
factor                           0
segment_actual_time              0
segment_osrm_time                0
segment_osrm_distance            0
segment_factor                   0
dtype: int64
```

There are 293 and 261 missing values in source_name,destination_name respectively.

In [10]:

```
data.nunique()
```

Out[10]:

```
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
```

Here data,route_type,is_cutoff are categorical type

conversion of categorical attributes to 'category'

In [11]:

```
data['data']=data['data'].astype('category')
data['route_type']=data['route_type'].astype('category')
data['is_cutoff']=data['is_cutoff'].astype('category')
```

In [12]:

```
data.dtypes
```

Out[12]:

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

- Lets check the valuecounts of categorical variables

In [13]:

```
d=data.select_dtypes(include='category')
a=list(d.columns)
a
```

Out[13]:

```
['data', 'route_type', 'is_cutoff']
```

In [14]:

```
for i in a:
    print(i)
    print(data[i].value_counts())
    print('\n')
```

```
data
training    104858
test         40009
Name: data, dtype: int64


route_type
FTL        99660
Carting    45207
Name: route_type, dtype: int64


is_cutoff
True     118749
False     26118
Name: is_cutoff, dtype: int64
```

Visual Analysis

Univariate Analysis of Continuous variables

In [15]:

```
con=list((data.select_dtypes(include=['int64','float64'])))
con
```

Out[15]:

```
['start_scan_to_end_scan',
 'cutoff_factor',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'factor',
 'segment_actual_time',
 'segment_osrm_time',
 'segment_osrm_distance',
 'segment_factor']
```

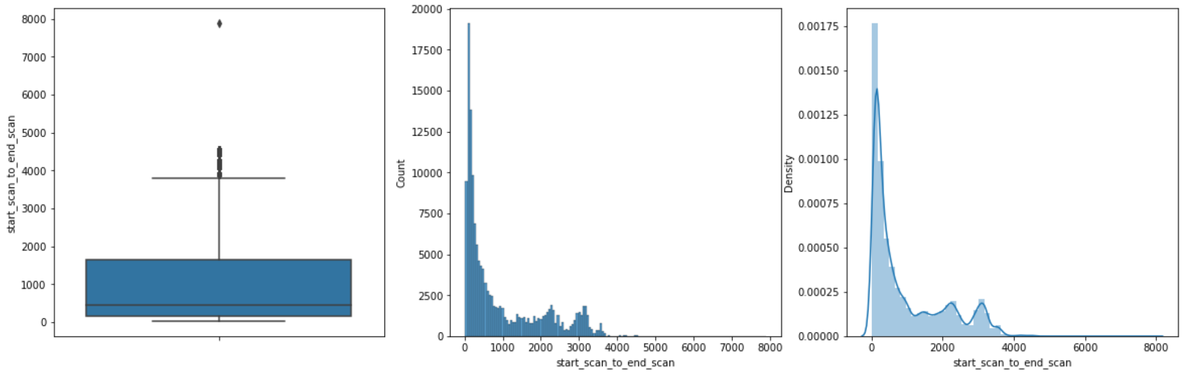
In [16]:

```
#start_scan_to_end_scan
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="start_scan_to_end_scan")
plt.subplot(1,3,2)
sns.histplot(data=data,x="start_scan_to_end_scan")
plt.subplot(1,3,3)
sns.distplot(a=data['start_scan_to_end_scan'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[16]:

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



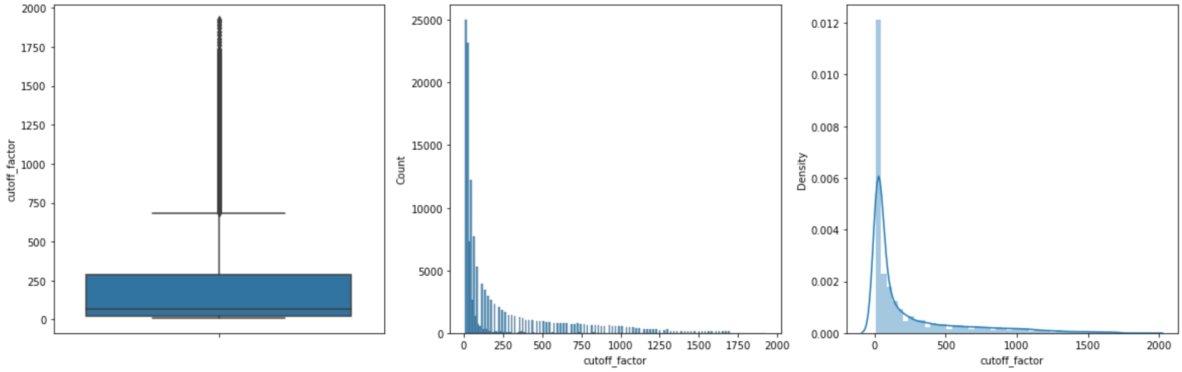
In [17]:

```
#cutoff_factor
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="cutoff_factor")
plt.subplot(1,3,2)
sns.histplot(data=data,x="cutoff_factor")
plt.subplot(1,3,3)
sns.distplot(a=data['cutoff_factor'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[17]:

```
<AxesSubplot:xlabel='cutoff_factor', ylabel='Density'>
```



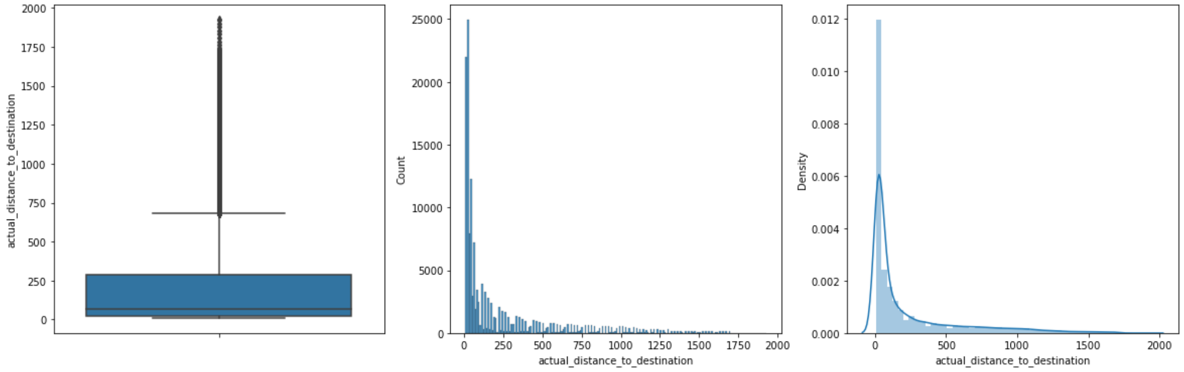
In [18]:

```
#actual_distance_to_destination
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="actual_distance_to_destination")
plt.subplot(1,3,2)
sns.histplot(data=data,x="actual_distance_to_destination")
plt.subplot(1,3,3)
sns.distplot(a=data['actual_distance_to_destination'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[18]:

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



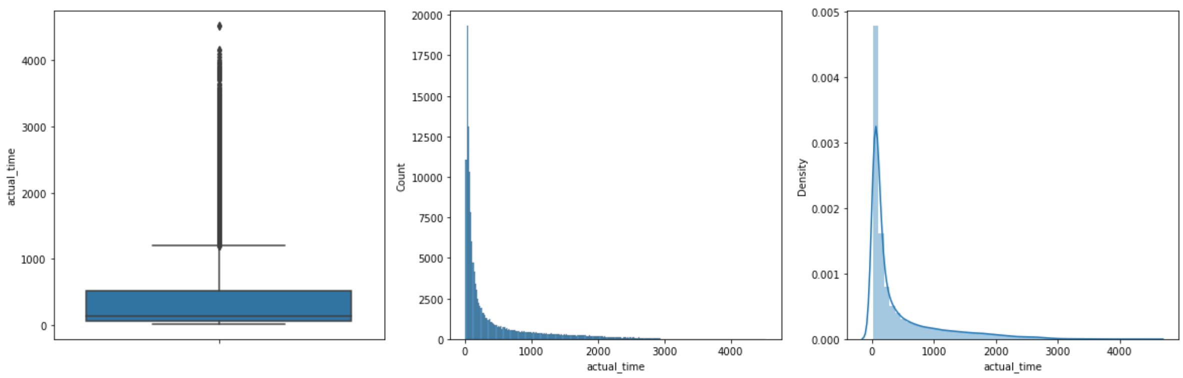
In [19]:

```
#actual_time
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="actual_time")
plt.subplot(1,3,2)
sns.histplot(data=data,x="actual_time")
plt.subplot(1,3,3)
sns.distplot(a=data['actual_time'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[19]:

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



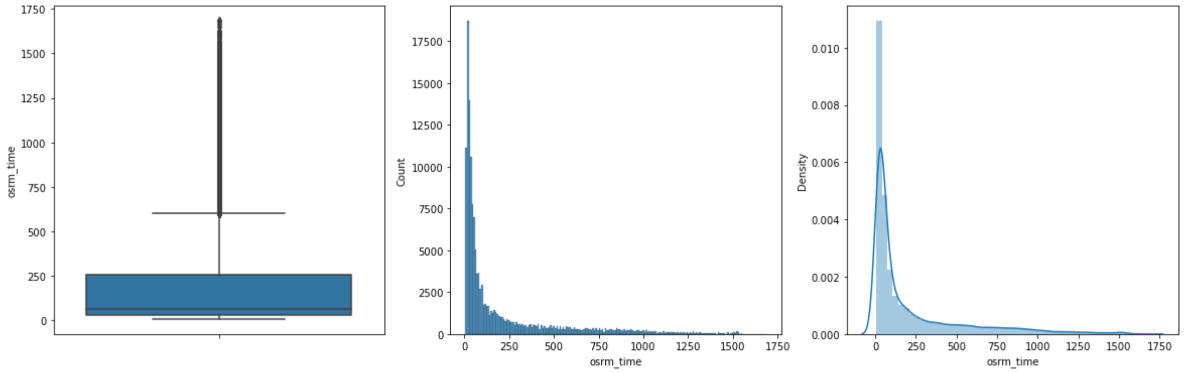
In [20]:

```
#osrm_time
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="osrm_time")
plt.subplot(1,3,2)
sns.histplot(data=data,x="osrm_time")
plt.subplot(1,3,3)
sns.distplot(a=data['osrm_time'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[20]:

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



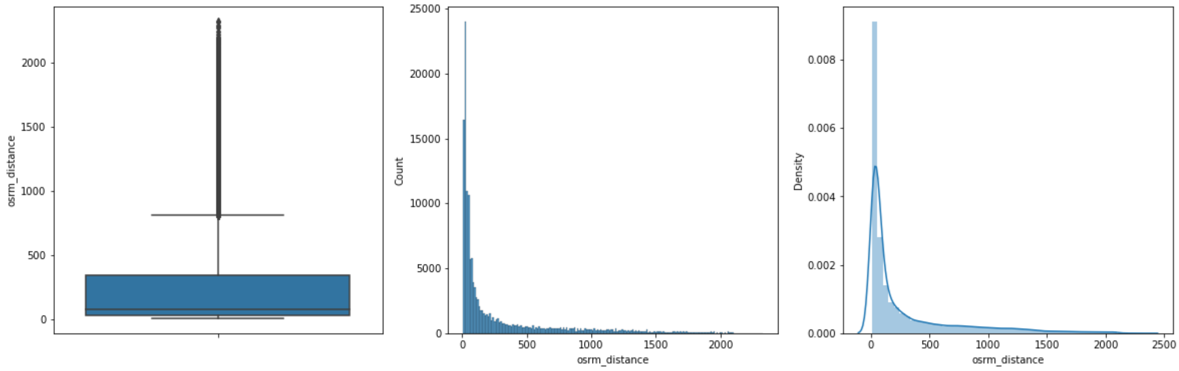
In [21]:

```
#osrm_distance
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="osrm_distance")
plt.subplot(1,3,2)
sns.histplot(data=data,x="osrm_distance")
plt.subplot(1,3,3)
sns.distplot(a=data['osrm_distance'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[21]:

```
<AxesSubplot:xlabel='osrm_distance', ylabel='Density'>
```



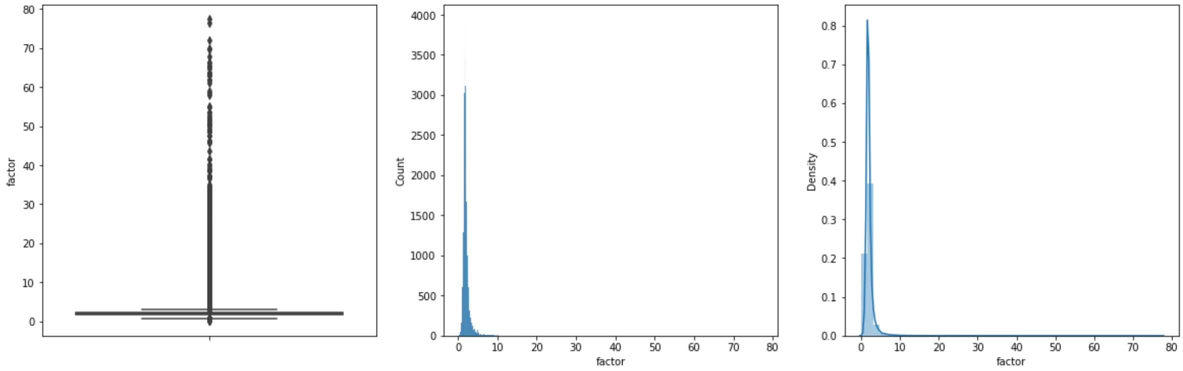
In [22]:

```
#factor
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="factor")
plt.subplot(1,3,2)
sns.histplot(data=data,x="factor")
plt.subplot(1,3,3)
sns.distplot(a=data['factor'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[22]:

```
<AxesSubplot:xlabel='factor', ylabel='Density'>
```



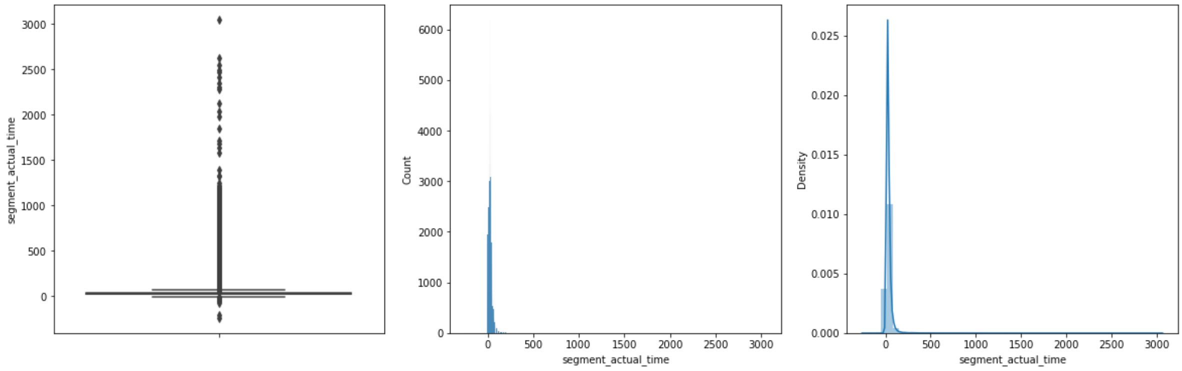
In [23]:

```
#segment_actual_time
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="segment_actual_time")
plt.subplot(1,3,2)
sns.histplot(data=data,x="segment_actual_time")
plt.subplot(1,3,3)
sns.distplot(a=data['segment_actual_time'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[23]:

```
<AxesSubplot:xlabel='segment_actual_time', ylabel='Density'>
```



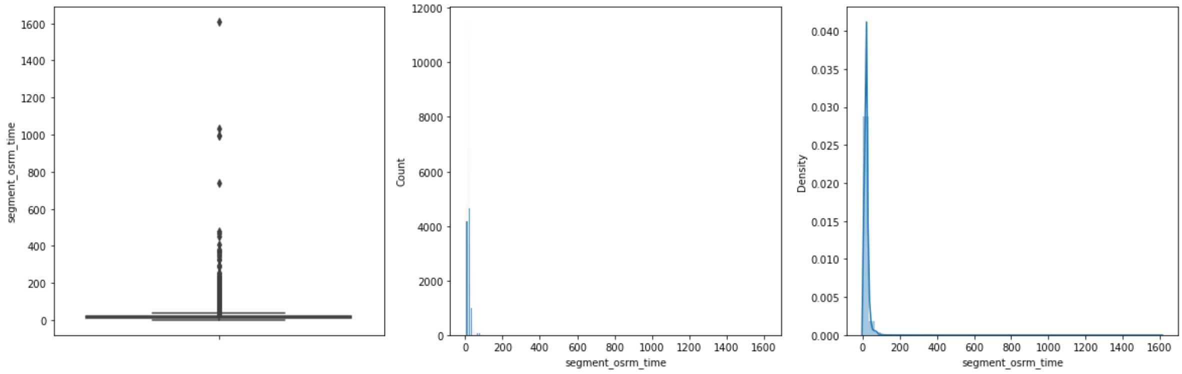
In [24]:

```
#segment_osrm_time
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="segment_osrm_time")
plt.subplot(1,3,2)
sns.histplot(data=data,x="segment_osrm_time")
plt.subplot(1,3,3)
sns.distplot(a=data['segment_osrm_time'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[24]:

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



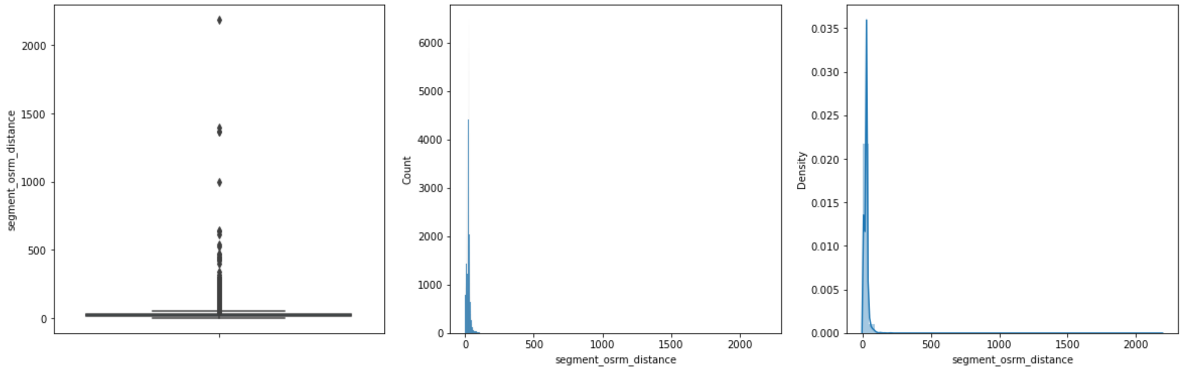
In [25]:

```
#segment_osrm_distance
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="segment_osrm_distance")
plt.subplot(1,3,2)
sns.histplot(data=data,x="segment_osrm_distance")
plt.subplot(1,3,3)
sns.distplot(a=data['segment_osrm_distance'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[25]:

```
<AxesSubplot:xlabel='segment_osrm_distance', ylabel='Density'>
```



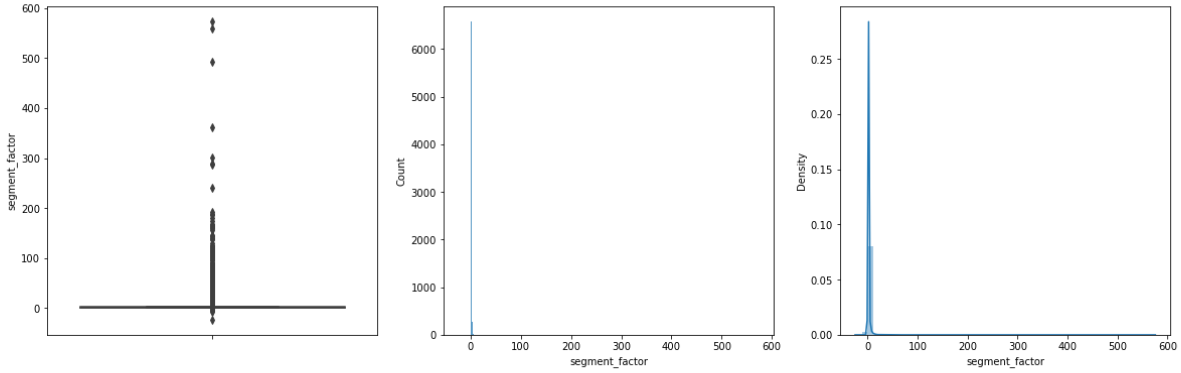
In [26]:

```
#segment_factor
plt.figure(figsize=(20,6))

plt.subplot(1,3,1)

sns.boxplot(data=data,y="segment_factor")
plt.subplot(1,3,2)
sns.histplot(data=data,x="segment_factor")
plt.subplot(1,3,3)
sns.distplot(a=data['segment_factor'],kde=True)
```

```
C:\Users\Ajith\anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[26]:

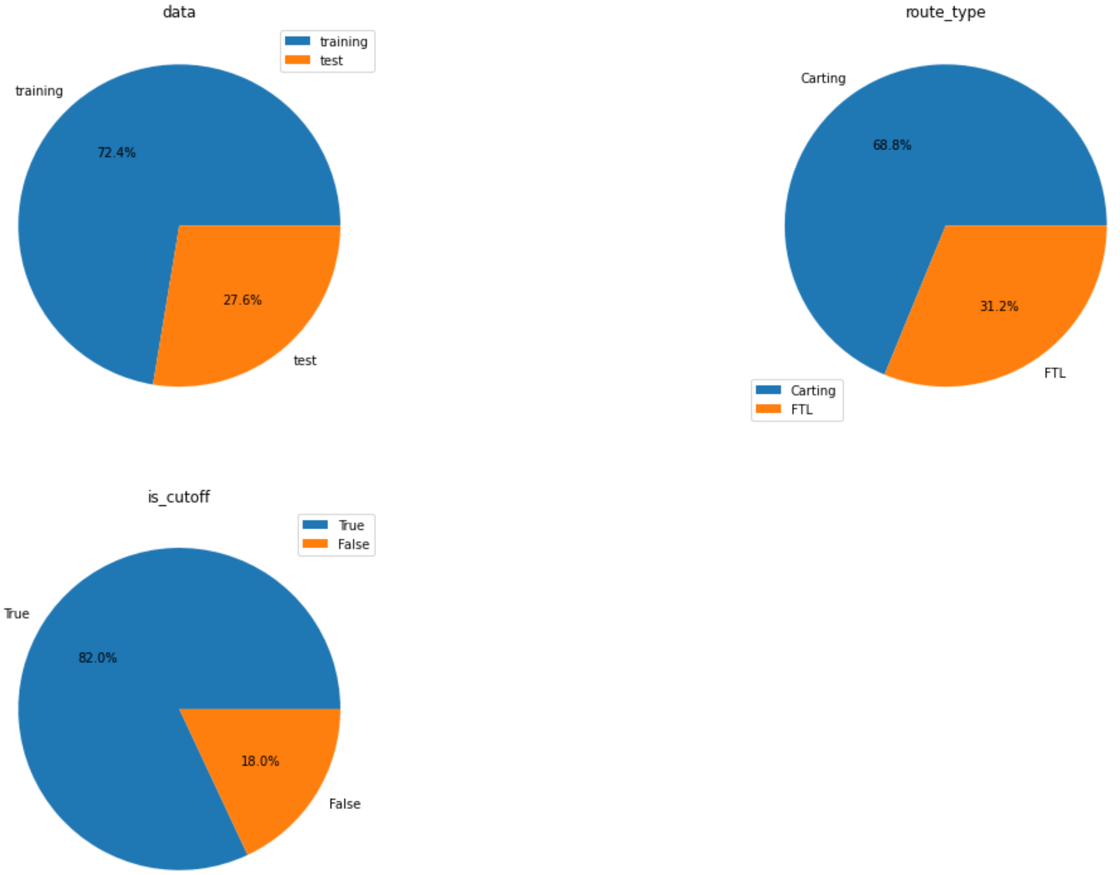
```
<AxesSubplot:xlabel='segment_factor', ylabel='Density'>
```



Univariate analysis of categorical variables

In [27]:

```
plt.figure(figsize = (20,20))
cols = ['data', 'route_type', 'is_cutoff']
k = 0
for i in cols:
    plt.subplot(321 + k)
    plt.pie(data[i].value_counts(), autopct = '%1.1f%%',
    labels = data[i].unique())
    plt.title(i)
    plt.legend()
    k += 1
plt.show()
```



Bivariate Analysis

In [28]:

```
con
```

Out[28]:

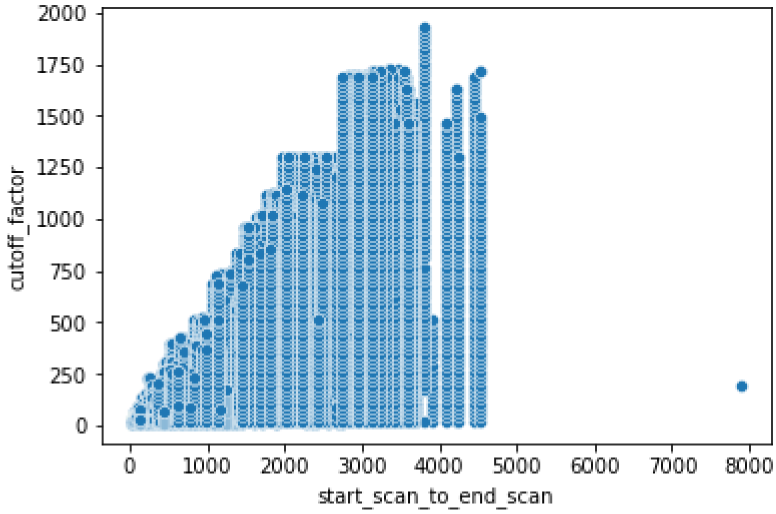
```
['start_scan_to_end_scan',
 'cutoff_factor',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'factor',
 'segment_actual_time',
 'segment_osrm_time',
 'segment_osrm_distance',
 'segment_factor']
```

In [29]:

```
sns.scatterplot(x='start_scan_to_end_scan',y='cutoff_factor',data=data)
```

Out[29]:

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

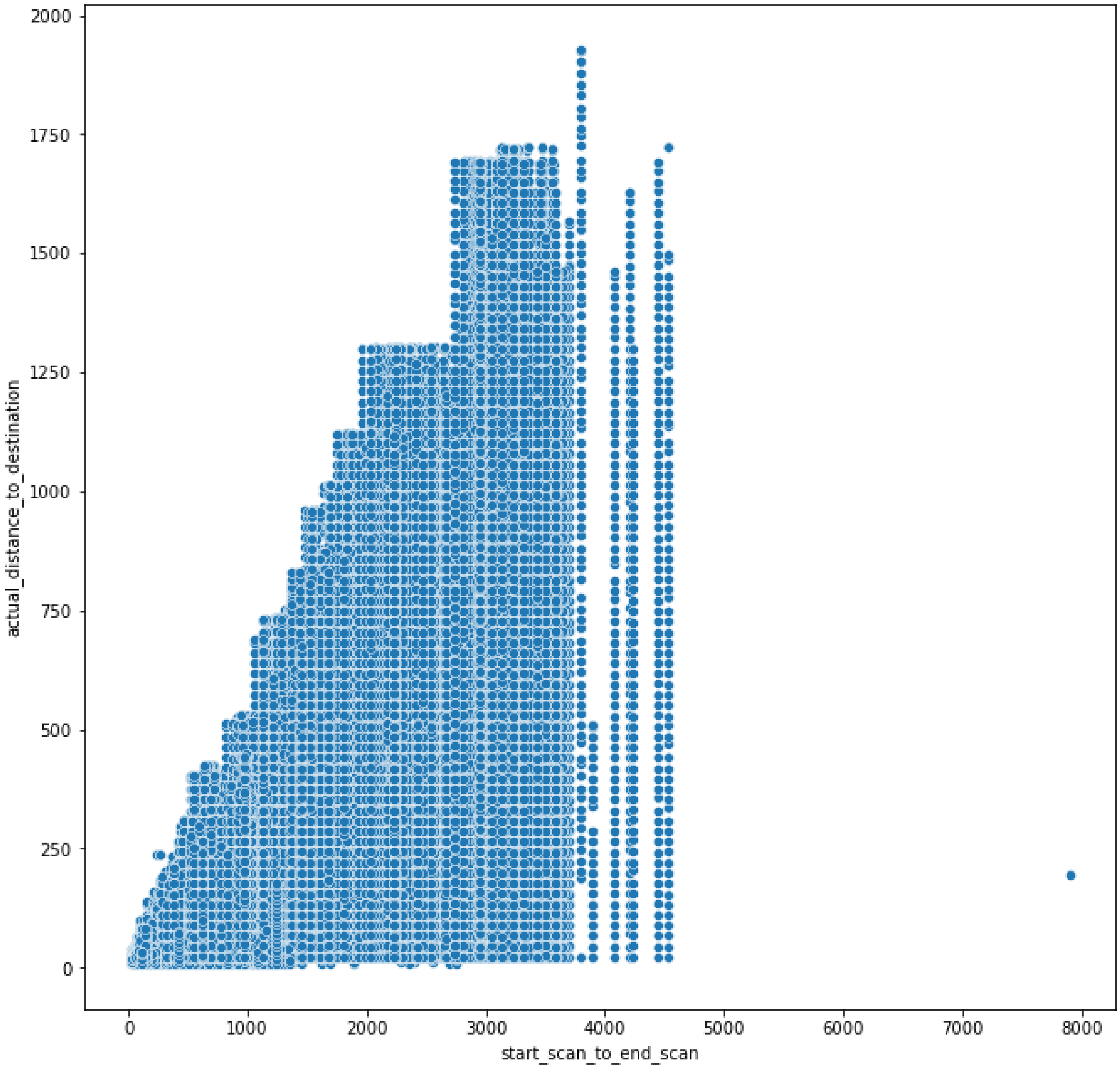


In [30]:

```
plt.figure(figsize=(11,11))
sns.scatterplot(x='start_scan_to_end_scan',y='actual_distance_to_destination',data=data)
```

Out[30]:

```
<AxesSubplot:xlabel='start_scan_to_end_scan', ylabel='actual_distance_to_des
tination'>
```

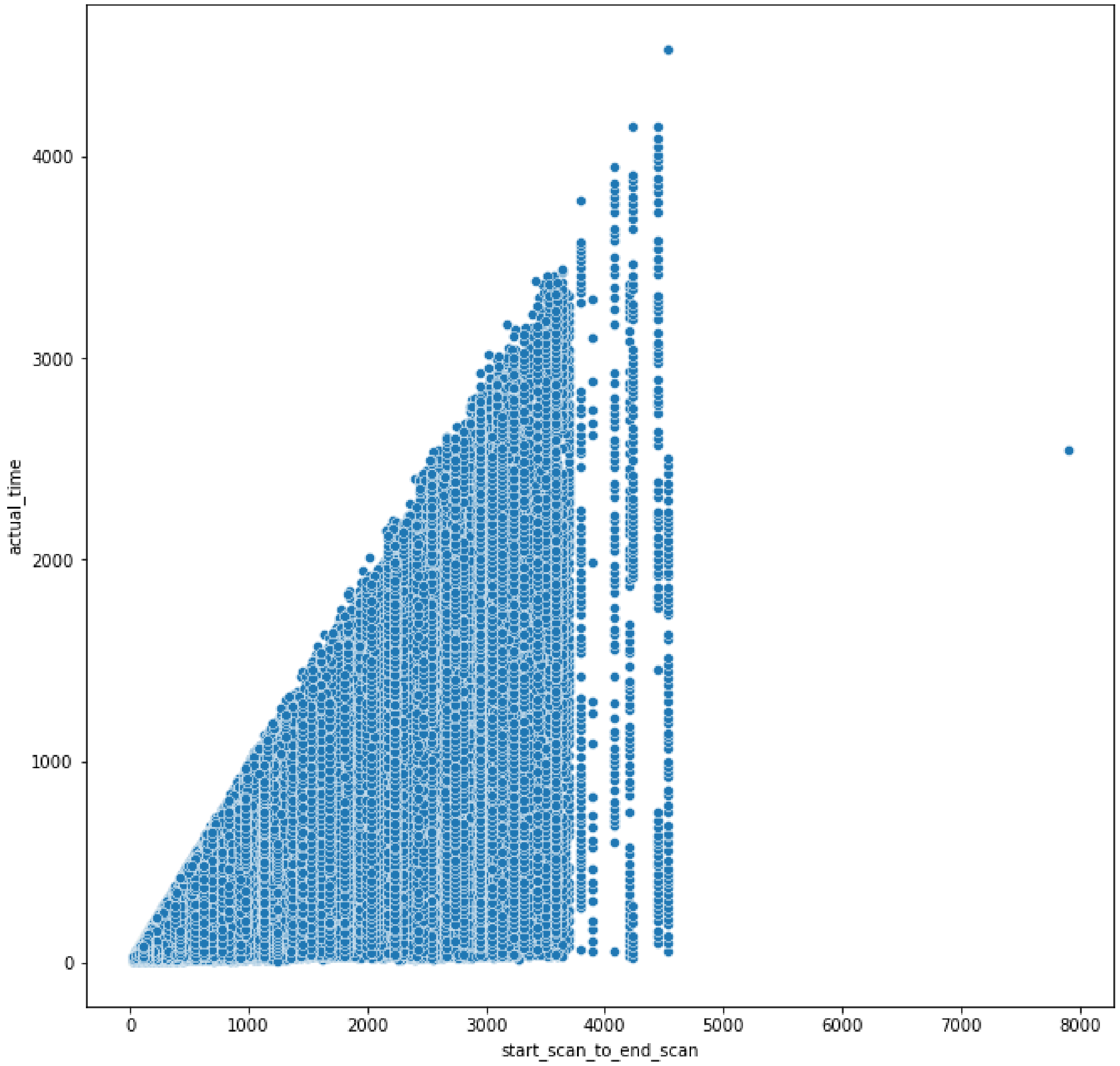


In [31]:

```
plt.figure(figsize=(11,11))
sns.scatterplot(x='start_scan_to_end_scan',y='actual_time',data=data)
```

Out[31]:

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

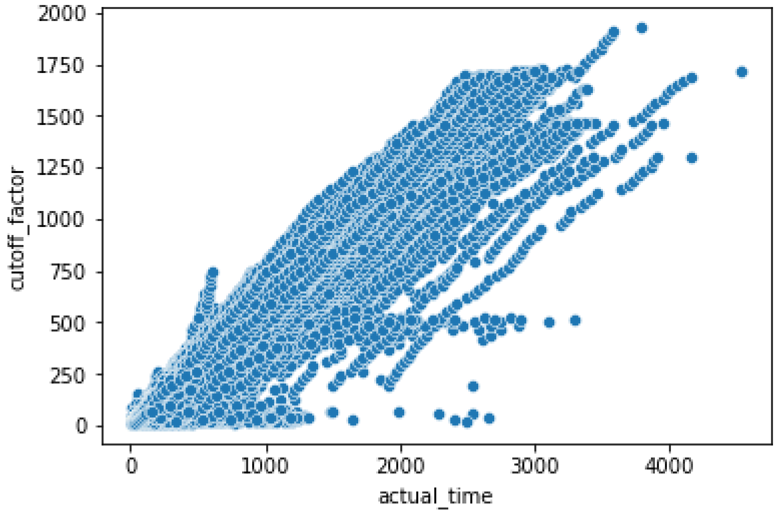


In [32]:

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

Out[32]:

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

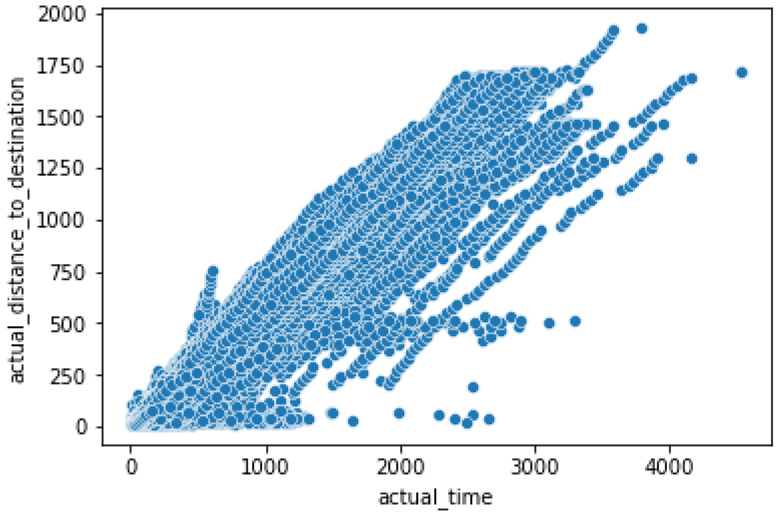


In [33]:

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

Out[33]:

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

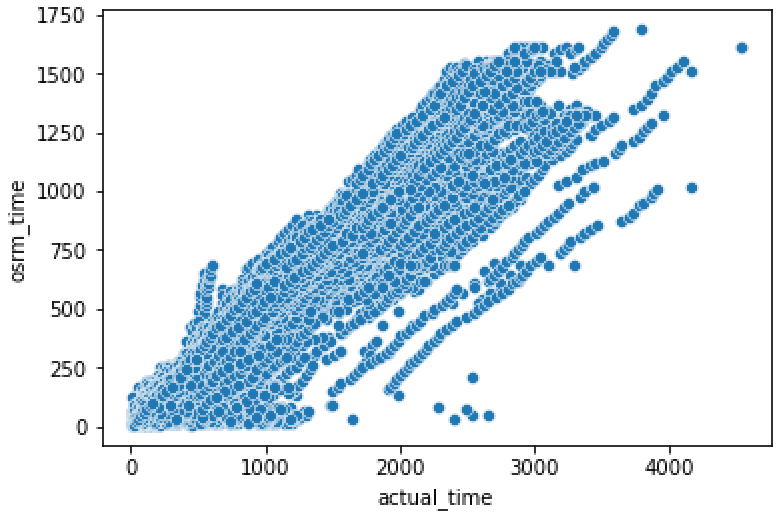


In [34]:

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

Out[34]:

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

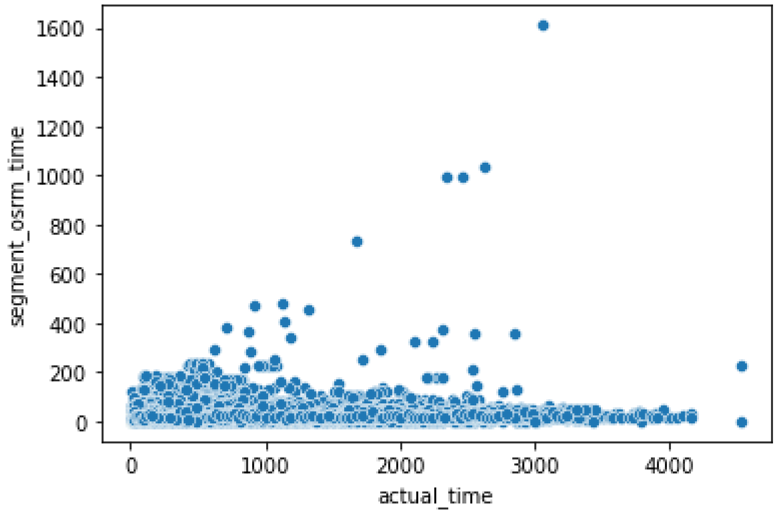


In [35]:

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

Out[35]:

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

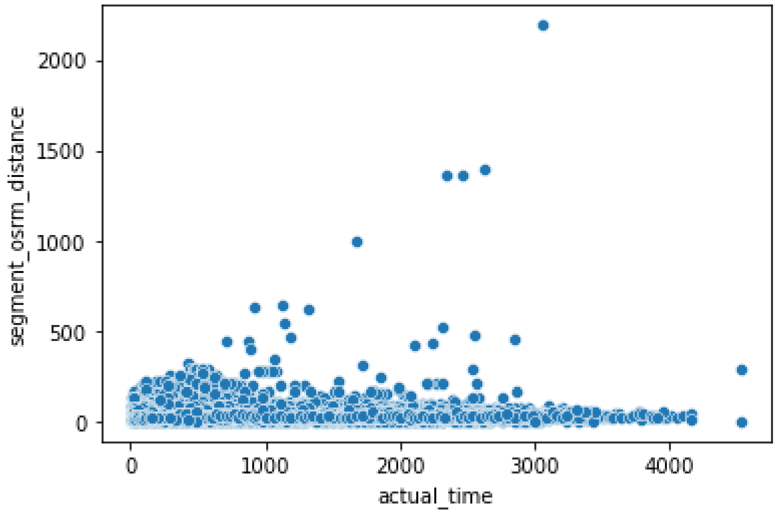


In [36]:

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

Out[36]:

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

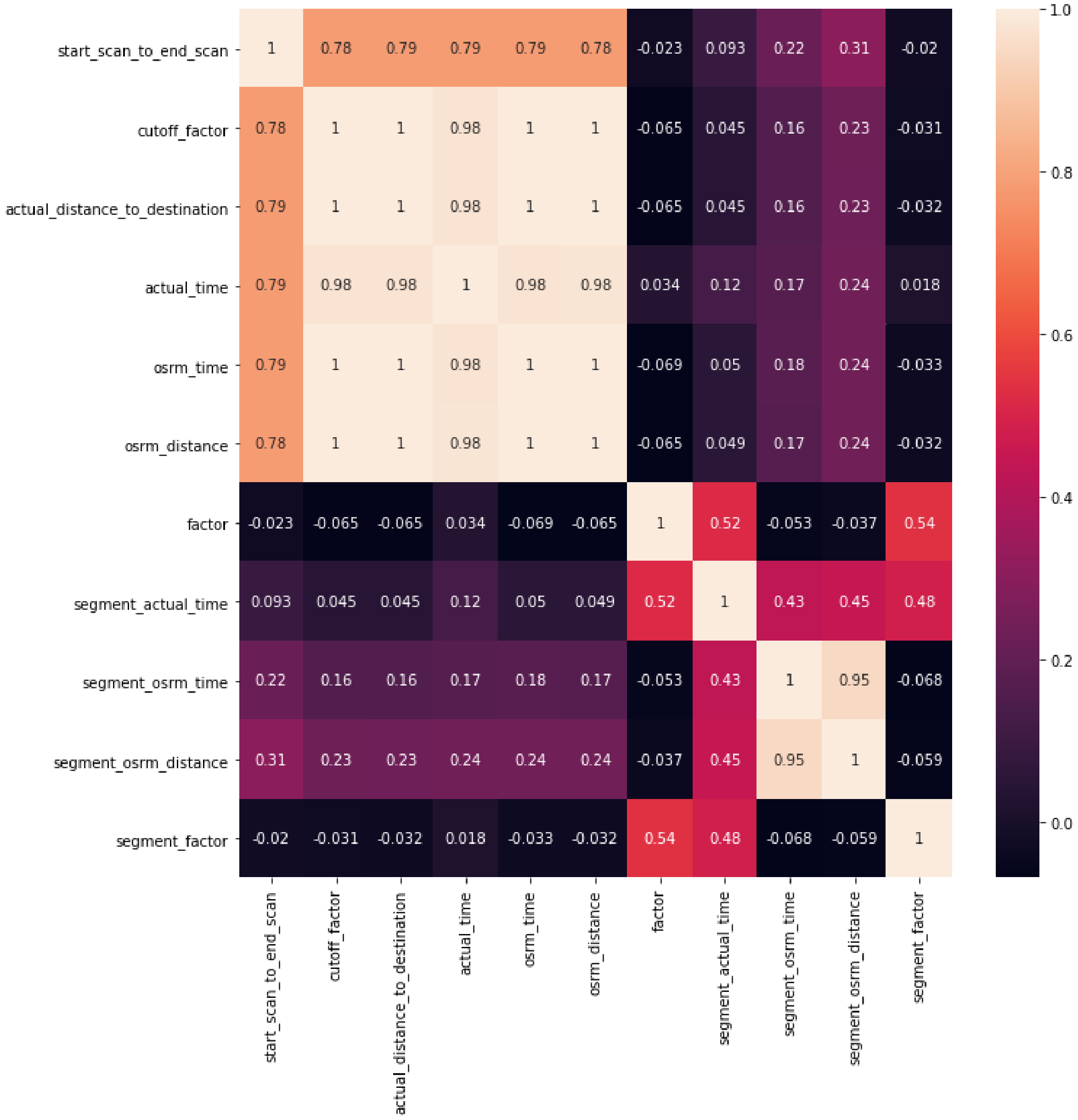


In [37]:

```
plt.figure(figsize=(11,11))
sns.heatmap(data.corr(),annot=True)
```

Out[37]:

```
<AxesSubplot:>
```



Observations

In [38]:

```
data['od_start_time']
```

Out[38]:

```
0         2018-09-20 03:21:32.418600
1         2018-09-20 03:21:32.418600
2         2018-09-20 03:21:32.418600
3         2018-09-20 03:21:32.418600
4         2018-09-20 03:21:32.418600
                     ...
144862    2018-09-20 16:24:28.436231
144863    2018-09-20 16:24:28.436231
144864    2018-09-20 16:24:28.436231
144865    2018-09-20 16:24:28.436231
144866    2018-09-20 16:24:28.436231
Name: od_start_time, Length: 144867, dtype: object
```

In [39]:

```
data['od_start_time']=pd.to_datetime(data['od_start_time'])
```

In [40]:

```
data['od_start_hour']=data['od_start_time'].dt.hour
data['od_start_min']=data['od_start_time'].dt.minute
```

In [41]:

```
data['od_start_hour']+=data['od_start_min']/60
```

In [42]:

```
data['od_start_hour']
```

Out[42]:

```
0          3.35
1          3.35
2          3.35
3          3.35
4          3.35
          ...
144862    16.40
144863    16.40
144864    16.40
144865    16.40
144866    16.40
Name: od_start_hour, Length: 144867, dtype: float64
```

In [43]:

```
data.drop(columns='od_start_min',axis=1,inplace=True)
```

In [44]:

```
data['od_end_time']
```

Out[44]:

```
0         2018-09-20 04:47:45.236797
1         2018-09-20 04:47:45.236797
2         2018-09-20 04:47:45.236797
3         2018-09-20 04:47:45.236797
4         2018-09-20 04:47:45.236797
                     ...
144862    2018-09-20 23:32:09.618069
144863    2018-09-20 23:32:09.618069
144864    2018-09-20 23:32:09.618069
144865    2018-09-20 23:32:09.618069
144866    2018-09-20 23:32:09.618069
Name: od_end_time, Length: 144867, dtype: object
```

In [45]:

```
data['od_end_time']=pd.to_datetime(data['od_end_time'])
```

In [46]:

```
data['od_end_hour']=data['od_end_time'].dt.hour
data['od_end_min']=data['od_end_time'].dt.minute
data['od_end_hour']+=data['od_end_min']/60
```

In [47]:

```
data.drop(columns='od_end_min',axis=1,inplace=True)
data
```

Out[47]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	so
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
...	...	...	...	...	...	
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN

144867 rows × 26 columns

In [48]:

```
data
```

Out[48]:

data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_cen
training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121A
training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121A
training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121A
training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121A
training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121A
...	...	...	...	...	
training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028A
training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028A
training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028A
training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028A
training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028A

rows × 26 columns

In [49]:

```
data['actual_time'].nunique()
```

Out[49]:

```
3182
```

In [50]:

```
x=data.groupby(by=['trip_uuid','source_center','destination_center'])
```

In [51]:

```
# Here we dont know what are is_cutoff ,cutoff_factor,cutoff_timestamp and segment_factor d
data.drop(columns=['is_cutoff' ,'cutoff_factor','cutoff_timestamp','segment_factor'],inplac
```

Feature Creation

Merging of rows and aggregation of fields

In [52]:

```
data['agg_actual_time']=data.groupby(by=['trip_uuid','source_center','destination_center'])
data['agg_segment_actual_time']=data.groupby(by=['trip_uuid','source_center','destination_c
data['agg_osrm_time']=data.groupby(by=['trip_uuid','source_center','destination_center'])['
data['agg_segment_osrm_time']=data.groupby(by=['trip_uuid','source_center','destination_cen
data['agg_osrm_distance']=data.groupby(by=['trip_uuid','source_center','destination_center'
data['agg_segment_osrm_distance']=data.groupby(by=['trip_uuid','source_center','destination
```

In [53]:

```
data
```

Out[53]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	so
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
...	...	...	...	...	...	
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND

144867 rows × 28 columns

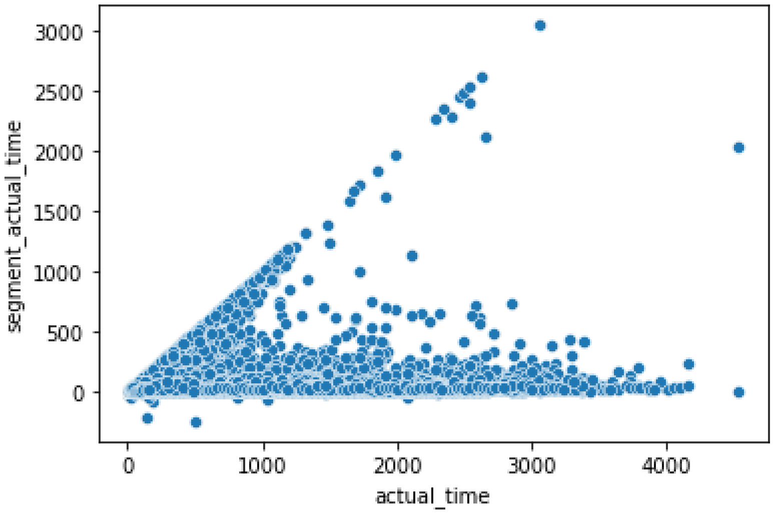
Comparison & Visualization of time and distance fields

In [54]:

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

Out[54]:

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

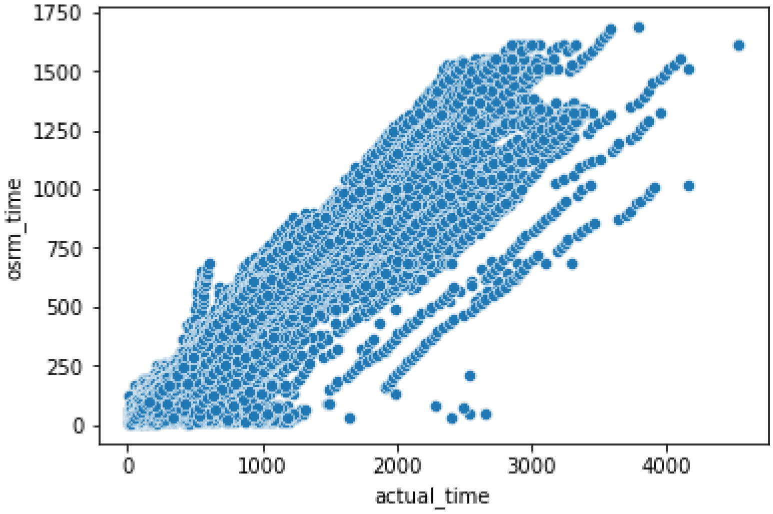


In [55]:

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

Out[55]:

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

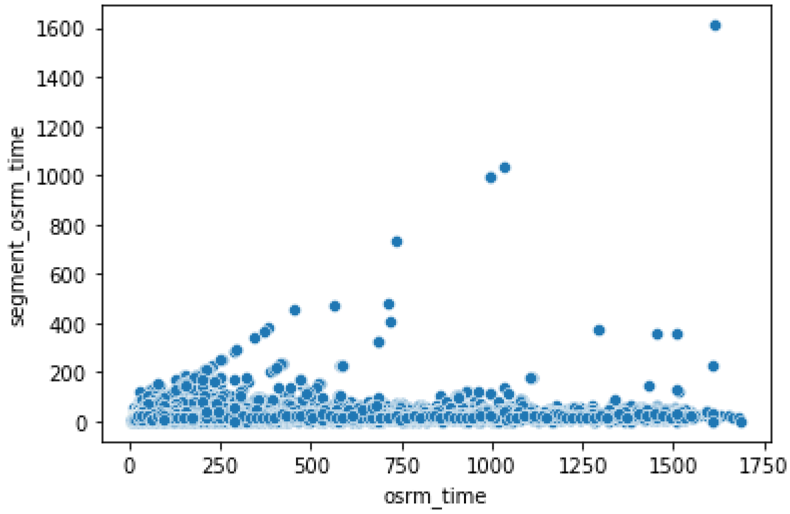


In [56]:

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

Out[56]:

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

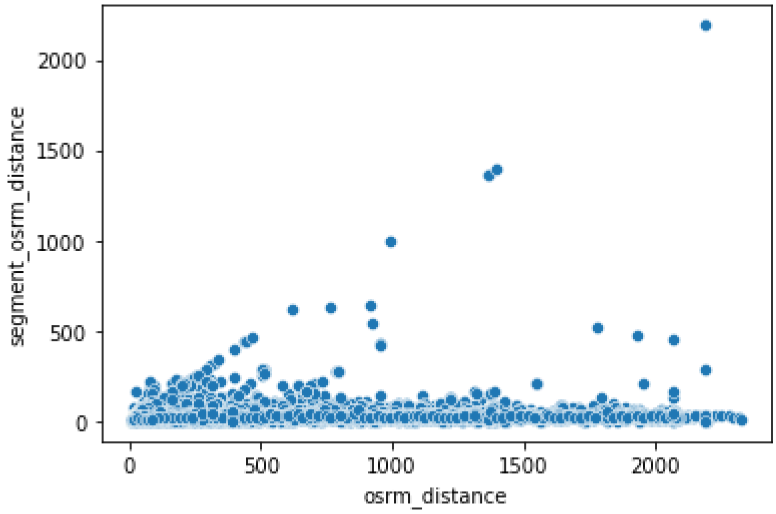


In [57]:

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

Out[57]:

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

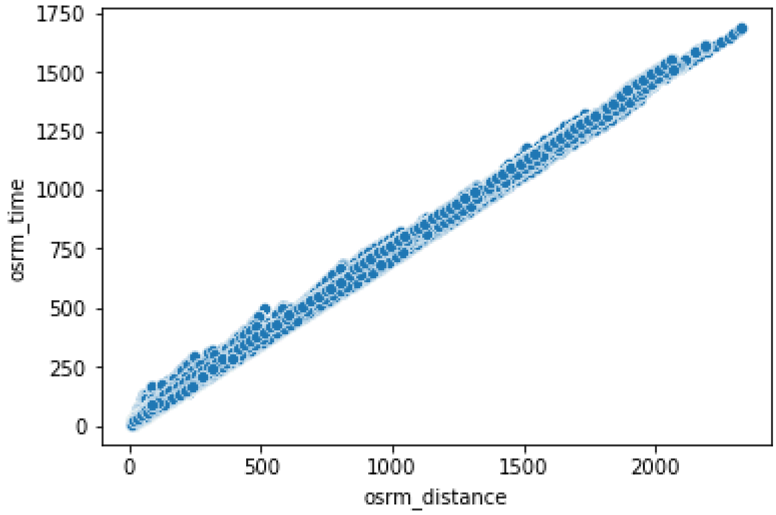


In [58]:

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

Out[58]:

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

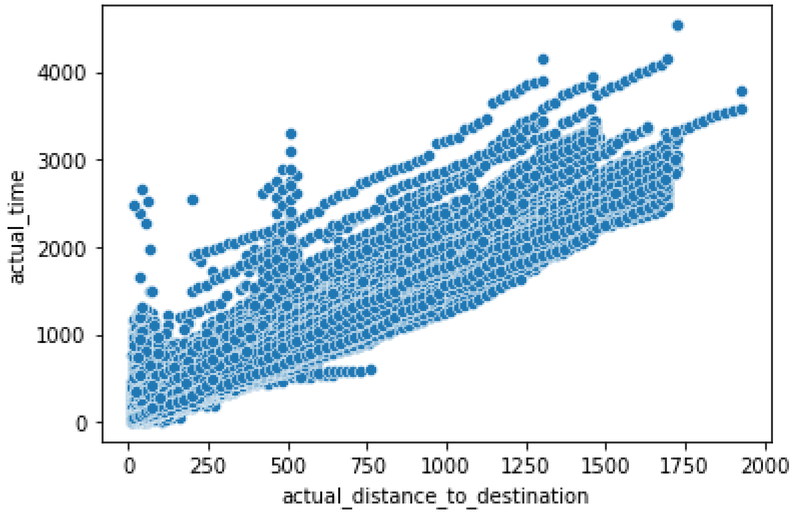


In [59]:

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

Out[59]:

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



Observations:

```
*The actual time and the estimated time are proportional to each other
* The graphs between actual distance and time is more deviated than the graph betw
een osrm_distance and time.
```

In [60]:

```
data['destination_name'].unique()
```

Out[60]:

```
array(['Khambhat_MotvdDPP_D (Gujarat)', 'Anand_Vaghasi_IP (Gujarat)',
       'Pune_Tathawde_H (Maharashtra)', ...,
       'Chennai_Mylapore (Tamil Nadu)', 'Naraingarh_Ward2DPP_D (Haryana)',
       'Mumbai_Ghansoli_DC (Maharashtra)'], dtype=object)
```

In [61]:

```
data['source_name'].unique()
```

Out[61]:

```
array(['Anand_VUNagar_DC (Gujarat)', 'Khambhat_MotvdDPP_D (Gujarat)',
       'Bhiwandi_Mankoli_HB (Maharashtra)', ...,
       'Dwarka_StnRoad_DC (Gujarat)', 'Bengaluru_Nelmngla_L (Karnataka)',
       'Kulithalai_AnnaNGR_D (Tamil Nadu)'], dtype=object)
```

In [62]:

```
data['source_name'].str.split('_')
```

Out[62]:

```
0          [Anand, VUNagar, DC (Gujarat)]
1          [Anand, VUNagar, DC (Gujarat)]
2          [Anand, VUNagar, DC (Gujarat)]
3          [Anand, VUNagar, DC (Gujarat)]
4          [Anand, VUNagar, DC (Gujarat)]
                       ...
144862    [Sonipat, Kundli, H (Haryana)]
144863    [Sonipat, Kundli, H (Haryana)]
144864    [Sonipat, Kundli, H (Haryana)]
144865    [Sonipat, Kundli, H (Haryana)]
144866    [Sonipat, Kundli, H (Haryana)]
Name: source_name, Length: 144867, dtype: object
```

In [63]:

```
data['source_city']=data['source_name'].apply(lambda x:str(x).split('_')[0])

data
```

Out[63]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	so
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IN
...	...	...	...	...	...	
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IN

144867 rows × 29 columns

In [85]:

```
data['source_place']=data['source_name'].apply(lambda x:str(x).split('_')[1:2])
data['source_place']=data['source_place'].apply(lambda x:str(x)[1:-1])
data['source_code']=data['source_name'].apply(lambda x:str(x).split('_')[2:3])
data
```

Out[85]:

gment_osrm_time	agg_osrm_distance	agg_segment_osrm_distance	source_city	source_place	so
11.0	11.9653	11.9653	Anand	'VUNagar'	[DC
20.0	33.6896	21.7243	Anand	'VUNagar'	[DC
27.0	66.2291	32.5395	Anand	'VUNagar'	[DC
39.0	111.7911	45.5619	Anand	'VUNagar'	[DC
44.0	166.0092	49.4772	Anand	'VUNagar'	[DC
...	...	...	...	...	
94.0	228.5453	65.3487	Sonipat	'Kundli'	[H
115.0	314.2282	82.7212	Sonipat	'Kundli'	[H
149.0	411.3215	103.4265	Sonipat	'Kundli'	[H
176.0	522.5924	122.3150	Sonipat	'Kundli'	[H
185.0	611.3243	131.1238	Sonipat	'Kundli'	[H

In [86]:

```
data['destination_city']=data['destination_name'].apply(lambda x:str(x).split('_')[0])
data['destination_place']=data['destination_name'].apply(lambda x:str(x).split('_')[1:2])
data['destination_place']=data['destination_place'].apply(lambda x:str(x)[1:-1])
data['destination_code']=data['destination_name'].apply(lambda x:str(x).split('_')[2:3])
data
```

Out[86]:

gment_osrm_time	agg_osrm_distance	agg_segment_osrm_distance	source_city	source_place	sc
11.0	11.9653	11.9653	Anand	'VUNagar'	[D
20.0	33.6896	21.7243	Anand	'VUNagar'	[D
27.0	66.2291	32.5395	Anand	'VUNagar'	[D
39.0	111.7911	45.5619	Anand	'VUNagar'	[D
44.0	166.0092	49.4772	Anand	'VUNagar'	[D
...	...	...	...	...	
94.0	228.5453	65.3487	Sonipat	'Kundli'	[H
115.0	314.2282	82.7212	Sonipat	'Kundli'	[H
149.0	411.3215	103.4265	Sonipat	'Kundli'	[H
176.0	522.5924	122.3150	Sonipat	'Kundli'	[H
185.0	611.3243	131.1238	Sonipat	'Kundli'	[H

In [87]:

```
data['trip_creation_time']
```

Out[87]:

```
0         2018-09-20 02:35:36.476840
1         2018-09-20 02:35:36.476840
2         2018-09-20 02:35:36.476840
3         2018-09-20 02:35:36.476840
4         2018-09-20 02:35:36.476840
                     ...
144862    2018-09-20 16:24:28.436231
144863    2018-09-20 16:24:28.436231
144864    2018-09-20 16:24:28.436231
144865    2018-09-20 16:24:28.436231
144866    2018-09-20 16:24:28.436231
Name: trip_creation_time, Length: 144867, dtype: object
```

In [88]:

```
data['trip_creation_time']=pd.to_datetime(data['trip_creation_time'])
```

In [89]:

```
data['trip_creation_time_month']=data['trip_creation_time'].dt.month_name()
data['trip_creation_time_year']=data['trip_creation_time'].dt.year
data['trip_creation_time_day']=data['trip_creation_time'].dt.day_name()
```

In [90]:

```
data
```

Out[90]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	so
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	INC
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	INC
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	INC
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	INC
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	INC
...	...	...	...	...	...	
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	INC
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	INC
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	INC
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	INC
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	INC

144867 rows × 37 columns

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 [91]:

```
data['diff_od_time_mins']=((data['od_end_time']-data['od_start_time']).astype('timedelta64[
```

In [92]:

```
data
```

Out[92]:

city	source_place	source_code	destination_city	destination_place	destination_code	trip_creatio
and	'VUNagar'	[DC (Gujarat)]	Khambhat	'MotvdDPP'	[D (Gujarat)]	
and	'VUNagar'	[DC (Gujarat)]	Khambhat	'MotvdDPP'	[D (Gujarat)]	
and	'VUNagar'	[DC (Gujarat)]	Khambhat	'MotvdDPP'	[D (Gujarat)]	
and	'VUNagar'	[DC (Gujarat)]	Khambhat	'MotvdDPP'	[D (Gujarat)]	
and	'VUNagar'	[DC (Gujarat)]	Khambhat	'MotvdDPP'	[D (Gujarat)]	
...	...	...	...	...	...	
ipat	'Kundli'	[H (Haryana)]	Gurgaon	'Bilaspur'	[HB (Haryana)]	
ipat	'Kundli'	[H (Haryana)]	Gurgaon	'Bilaspur'	[HB (Haryana)]	
ipat	'Kundli'	[H (Haryana)]	Gurgaon	'Bilaspur'	[HB (Haryana)]	
ipat	'Kundli'	[H (Haryana)]	Gurgaon	'Bilaspur'	[HB (Haryana)]	
ipat	'Kundli'	[H (Haryana)]	Gurgaon	'Bilaspur'	[HB (Haryana)]	

In [ ]:

In [94]:

hypothesis testing/ visual analysis between actual_time aggregated value and OSRM time aggregated value

In [95]:

```
# H0:The average of actual and predicted times are equal
# Ha: The average of actual and predicted times are not equal
ztest(data['agg_actual_time'],data['agg_osrm_time'])
```

Out[95]:

```
(66.07555350813415, 0.0)
```

*Here p value is less than alpha so we reject null hypothesis

Hypothesis testing between actual_time aggregated value and segment actual time aggregated value

In [96]:

```
# H0:The average of actual and segmented actual times are equal
# Ha: The average of actual and segmented actual times are not equal
ztest(data['agg_actual_time'],data['agg_segment_actual_time'])
```

Out[96]:

```
(149.45336128899106, 0.0)
```

Do hypothesis testing/ visual analysis between osrm distance aggregated value and segment osrm distance aggregated value

In [97]:

```
# H0:The average of osrm distance and segmented osrm distance are equal
# Ha: The average of osrm distance and segmented osrm distance are not equal
ztest(data['agg_osrm_distance'],data['agg_segment_osrm_distance'])
```

Out[97]:

```
(146.87853467796583, 0.0)
```

Do hypothesis testing/ visual analysis between osrm time aggregated value and segment osrm time aggregated value

In [98]:

```
# H0:The average of osrm time and segmented osrm time are equal
# Ha: The average of osrm time and segmented osrm time are not equal
ztest(data['agg_osrm_time'],data['agg_segment_osrm_time'])
```

Out[98]:

```
(147.37782612875128, 0.0)
```

Here we are rejecting all our null hypothesis because our features data is not following normal distributions

OUTLIER TREATMENT

In [99]:

```
con=list((data.select_dtypes(include=['int64','float64'])))
con
```

Out[99]:

```
['start_scan_to_end_scan',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'factor',
 'segment_actual_time',
 'segment_osrm_time',
 'segment_osrm_distance',
 'od_start_hour',
 'od_end_hour',
 'agg_actual_time',
 'agg_segment_actual_time',
 'agg_osrm_time',
 'agg_segment_osrm_time',
 'agg_osrm_distance',
 'agg_segment_osrm_distance',
 'trip_creation_time_year',
 'diff_od_time_mins']
```

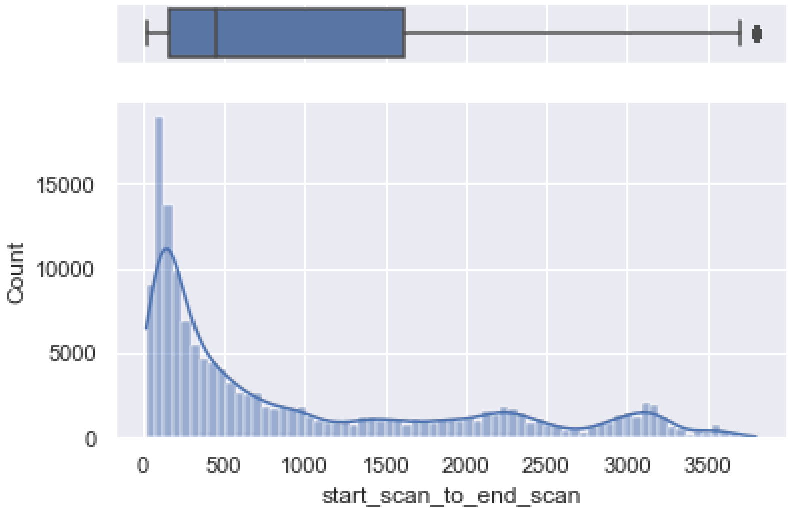
In [100]:

```
#start_scan_to_end_scan
Q3 = data['start_scan_to_end_scan'].quantile(0.75)
Q1 = data['start_scan_to_end_scan'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['start_scan_to_end_scan']>lower) & (data['start_scan_to_end_scan']<upper)
```

```
3843.5 -2048.5
```

In [101]:

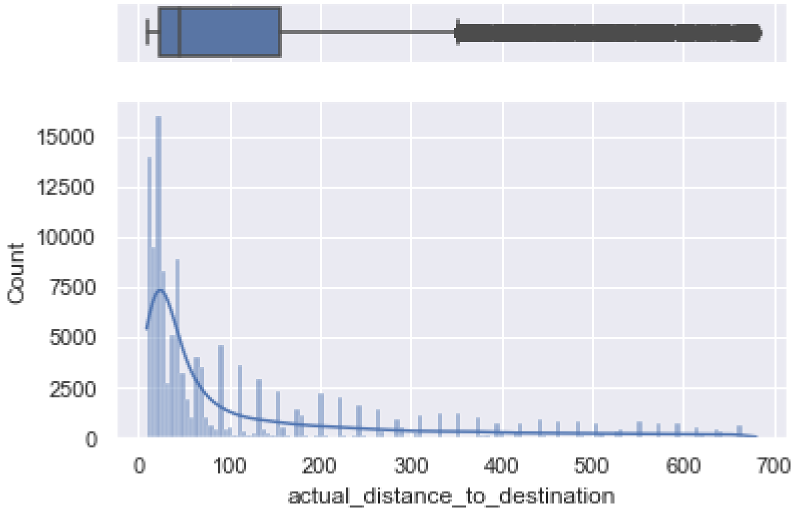
```
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='start_scan_to_end_scan', ax=ax_box)
sns.histplot(data=data, x='start_scan_to_end_scan', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```



In [103]:

```
#actual_distance_to_destination
Q3 = data['actual_distance_to_destination'].quantile(0.75)
Q1 = data['actual_distance_to_destination'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['actual_distance_to_destination']>lower) & (data['actual_distance_to_dest
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='actual_distance_to_destination', ax=ax_box)
sns.histplot(data=data, x='actual_distance_to_destination', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

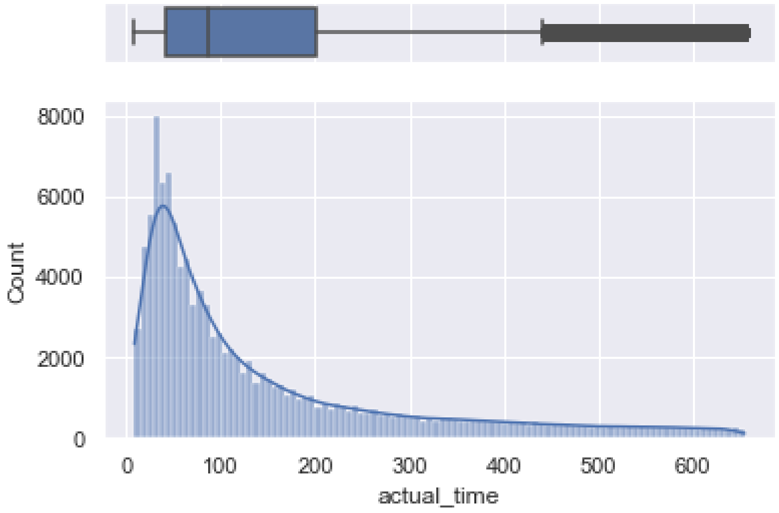
```
681.0150207887134 -371.27176693600586
```



In [104]:

```
#actual_time
Q3 = data['actual_time'].quantile(0.75)
Q1 = data['actual_time'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['actual_time']>lower) & (data['actual_time']<upper)]
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='actual_time', ax=ax_box)
sns.histplot(data=data, x='actual_time', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

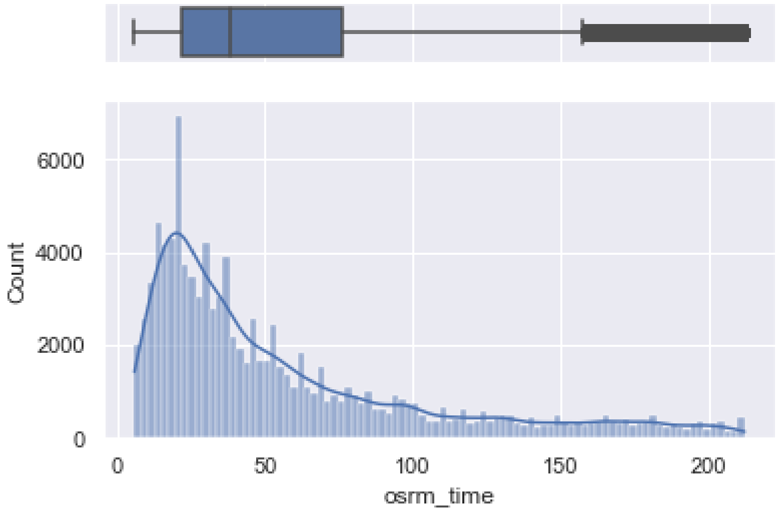
```
656.0 -320.0
```



In [106]:

```
#osrm_time
Q3 = data['osrm_time'].quantile(0.75)
Q1 = data['osrm_time'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['osrm_time']>lower) & (data['osrm_time']<upper)]
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='osrm_time', ax=ax_box)
sns.histplot(data=data, x='osrm_time', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

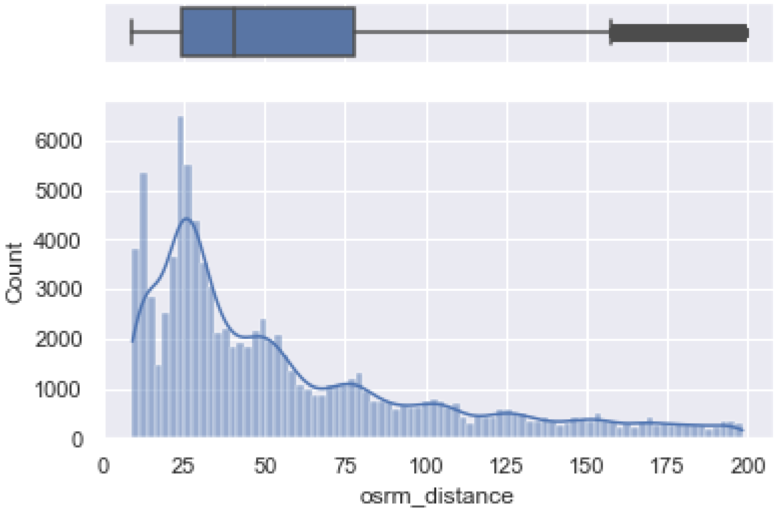
```
213.0 -91.0
```



In [107]:

```
#osrm_distance
Q3 = data['osrm_distance'].quantile(0.75)
Q1 = data['osrm_distance'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['osrm_distance']>lower) & (data['osrm_distance']<upper)]
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='osrm_distance', ax=ax_box)
sns.histplot(data=data, x='osrm_distance', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

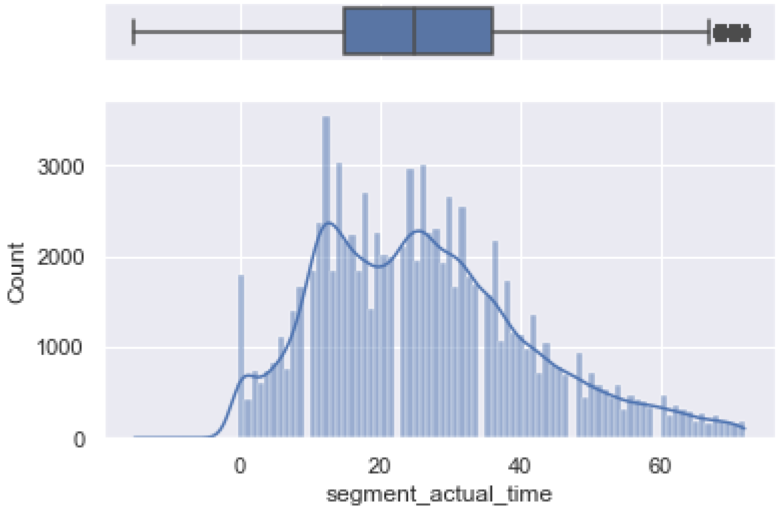
```
198.37162499999997 -78.85917499999996
```



In [108]:

```
#segment_actual_time
Q3 = data['segment_actual_time'].quantile(0.75)
Q1 = data['segment_actual_time'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['segment_actual_time']>lower) & (data['segment_actual_time']<upper)]
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='segment_actual_time', ax=ax_box)
sns.histplot(data=data, x='segment_actual_time', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

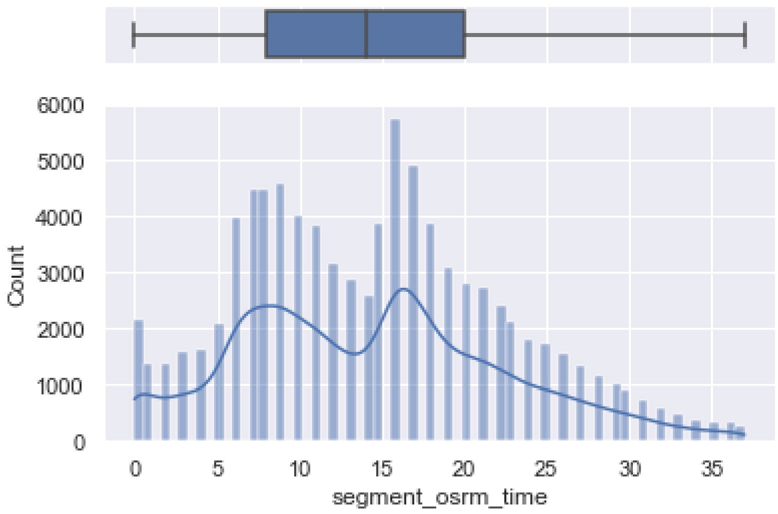
```
72.5 -19.5
```



In [109]:

```
#segment_osrm_time
Q3 = data['segment_osrm_time'].quantile(0.75)
Q1 = data['segment_osrm_time'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['segment_osrm_time']>lower) & (data['segment_osrm_time']<upper)]
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='segment_osrm_time', ax=ax_box)
sns.histplot(data=data, x='segment_osrm_time', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

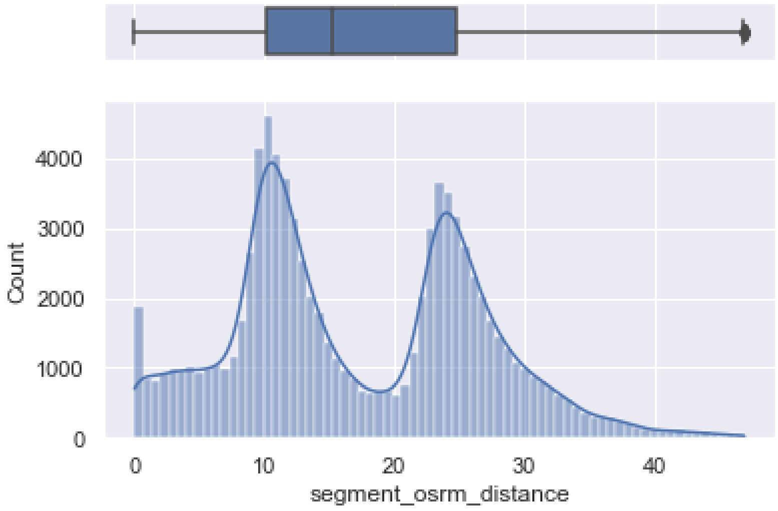
```
38.0 -10.0
```



In [110]:

```
#segment_osrm_distance
Q3 = data['segment_osrm_distance'].quantile(0.75)
Q1 = data['segment_osrm_distance'].quantile(0.25)
IQR = Q3-Q1
upper = Q3+(1.5*IQR)
lower = Q1-(1.5*IQR)
print(upper,lower)
data = data[(data['segment_osrm_distance']>lower) & (data['segment_osrm_distance']<upper)]
sns.set(style="darkgrid")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85
sns.boxplot(data=data,x='segment_osrm_distance', ax=ax_box)
sns.histplot(data=data, x='segment_osrm_distance', ax=ax_hist,kde=True)
ax_box.set(xlabel='')
plt.show()
```

```
46.97781250000001 -12.066087500000007
```



SUMMARY

- Above i have done the outlier detection and correction

One hot Encoding

In [113]:

```
from sklearn.preprocessing import OneHotEncoder
```

In [115]:

```
data['route_type']
```

Out[115]:

```
0         Carting
1         Carting
2         Carting
3         Carting
4         Carting
           ...   
144861    Carting
144862    Carting
144863    Carting
144864    Carting
144865    Carting
Name: route_type, Length: 87891, dtype: category
Categories (2, object): ['Carting', 'FTL']
```

In [117]:

```
one=OneHotEncoder()
one_data=pd.DataFrame(one.fit_transform(data[['route_type']]).toarray())
data=data.join(one_data)
```

In [118]:

```
data
```

Out[118]:

trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_center	s
2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_
2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_
2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_
2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_
2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_
...	...	...	...	...	
2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Son
2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Son
2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Son
2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Son
2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Son

olumns

In [126]:

```
data_con=data[['start_scan_to_end_scan',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'segment_actual_time',
 'segment_osrm_time',
 'segment_osrm_distance']]
from sklearn.preprocessing import MinMaxScaler

norm = MinMaxScaler().fit(data_con)

data_norm_con = norm.transform(data_con)
data_norm_con
```

Out[126]:

```
array([[0.01745108, 0.00818264, 0.00773994, ..., 0.33333333, 0.2972973 ,
        0.25475484],
       [0.01745108, 0.05663722, 0.02321981, ..., 0.28735632, 0.24324324,
        0.20778021],
       [0.01745108, 0.1062275 , 0.04798762, ..., 0.35632184, 0.18918919,
        0.23026791],
       ...,
       [0.10761502, 0.25701573, 0.17182663, ..., 0.47126437, 0.56756757,
        0.36988028],
       [0.10761502, 0.32581771, 0.20278638, ..., 0.40229885, 0.91891892,
        0.44083938],
       [0.10761502, 0.36866314, 0.23065015, ..., 0.36781609, 0.72972973,
        0.40215764]])
```

In [ ]: