In [1]:

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder
from sklearn.utils import resample
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
df = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/551/original/delhivery_data.csv?
```

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

Out[4]:

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

5 rows × 24 columns

In [5]:
```
df.shape
```

Out[5]:
```
(144867, 24)
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['data', 'trip_creation_time', 'route_schedule_uuid', 'route_type',
       'trip_uuid', 'source_center', 'source_name', 'destination_center',
       'destination_name', 'od_start_time', 'od_end_time',
       'start_scan_to_end_scan', 'is_cutoff', 'cutoff_factor',
       'cutoff_timestamp', 'actual_distance_to_destination', 'actual_time',
       'osrm_time', 'osrm_distance', 'factor', 'segment_actual_time',
       'segment_osrm_time', 'segment_osrm_distance', 'segment_factor'],
      dtype='object')
```

In [7]:
```
df.dtypes
```

Out[7]:

	0
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

	0
segment_osrm_distance	float64
segment_factor	float64

dtype: object

In [8]:
```
df.isnull().sum()
```

Out[8]:

	0
data	0
trip_creation_time	0
route_schedule_uuid	0
route_type	0
trip_uuid	0
source_center	0
source_name	293
destination_center	0
destination_name	261
od_start_time	0
od_end_time	0
start_scan_to_end_scan	0
is_cutoff	0
cutoff_factor	0
cutoff_timestamp	0
actual_distance_to_destination	0
actual_time	0
osrm_time	0
osrm_distance	0
factor	0
segment_actual_time	0
segment_osrm_time	0

	0
segment_osrm_distance	0
segment_factor	0

dtype: int64

In [9]:
```
df.describe()
```

Out[9]:

	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	actual_time	osrm_time	osrm_distance	
count	144867.000000	144867.000000	144867.000000	144867.000000	144867.000000	144867.000000	14486
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	7

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144867 entries, 0 to 144866
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   data                            144867 non-null  object 
 1   trip_creation_time              144867 non-null  object 
 2   route_schedule_uuid             144867 non-null  object 
 3   route_type                      144867 non-null  object 
 4   trip_uuid                       144867 non-null  object 
 5   source_center                   144867 non-null  object 
 6   source_name                     144574 non-null  object 
 7   destination_center              144867 non-null  object 
 8   destination_name                144606 non-null  object 
 9   od_start_time                   144867 non-null  object 
 10  od_end_time                     144867 non-null  object 
 11  start_scan_to_end_scan          144867 non-null  float64
 12  is_cutoff                       144867 non-null  bool   
 13  cutoff_factor                   144867 non-null  int64  
 14  cutoff_timestamp                144867 non-null  object 
 15  actual_distance_to_destination  144867 non-null  float64
 16  actual_time                     144867 non-null  float64
 17  osrm_time                       144867 non-null  float64
 18  osrm_distance                   144867 non-null  float64
 19  factor                          144867 non-null  float64
 20  segment_actual_time             144867 non-null  float64
 21  segment_osrm_time               144867 non-null  float64
 22  segment_osrm_distance           144867 non-null  float64
 23  segment_factor                  144867 non-null  float64
dtypes: bool(1), float64(10), int64(1), object(12)
memory usage: 25.6+ MB
```

In [11]:
```
df.describe(include='object')
```

Out[11]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_center	source_name
count	144867	144867	144867	144867	144867	144867	144574
unique	2	14817	1504	2	14817	1508	1498
top	training	2018-09-28 05:23:15.359220	thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069f...	FTL	trip-153811219535896559	IND000000ACB	Gurgaon_Bilaspur_HE (Haryana)
freq	104858	101	1812	99660	101	23347	23347

In [12]:
```
cat_columns=df.dtypes=='object'
category=list(cat_columns[cat_columns].index)
num_columns=df.dtypes!='object'
numerical=list(num_columns[num_columns].index)
```

In [13]:
```
category_data=df[category]
category_data
```

Out[13]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_center	source_name
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
...	...	...	...	...	...	...	...
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)

144867 rows × 12 columns

In [14]:
```
for col in category_data.columns:
    print(col,':',category_data[col].nunique())
```

```
data : 2
trip_creation_time : 14817
route_schedule_uuid : 1504
route_type : 2
trip_uuid : 14817
source_center : 1508
source_name : 1498
destination_center : 1481
destination_name : 1468
od_start_time : 26369
od_end_time : 26369
cutoff_timestamp : 93180
```

In [15]:
```
numerical_data=df[numerical]
numerical_data
```

Out[15]:

	start_scan_to_end_scan	is_cutoff	cutoff_factor	actual_distance_to_destination	actual_time	osrm_time	osrm_distance	
0	86.0	True	9	10.435660	14.0	11.0	11.9653	1
1	86.0	True	18	18.936842	24.0	20.0	21.7243	1
2	86.0	True	27	27.637279	40.0	28.0	32.5395	1
3	86.0	True	36	36.118028	62.0	40.0	45.5620	1
4	86.0	False	39	39.386040	68.0	44.0	54.2181	1
...	...	...	...	...	...	...	...	
144862	427.0	True	45	45.258278	94.0	60.0	67.9280	1
144863	427.0	True	54	54.092531	120.0	76.0	85.6829	1
144864	427.0	True	63	66.163591	140.0	88.0	97.0933	1
144865	427.0	True	72	73.680667	158.0	98.0	111.2709	1
144866	427.0	False	70	70.039010	426.0	95.0	88.7319	4

144867 rows × 12 columns

In [16]:
```
for col in numerical_data.columns:
    print(col,':',numerical_data[col].nunique())
```

```
start_scan_to_end_scan : 1915
is_cutoff : 2
cutoff_factor : 501
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
```

In [17]:
```
df['trip_creation_time']=pd.to_datetime(df['trip_creation_time'])
df['od_start_time']=pd.to_datetime(df['od_start_time'])
df['od_end_time']=pd.to_datetime(df['od_end_time'])
# data['cutoff_timestamp']=pd.to_datetime(data['cutoff_timestamp'])
```

In [18]:
```
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  datetime64[ns]
 2   route_schedule_uuid             144867 non-null  object
 3   route_type                      144867 non-null  object
 4   trip_uuid                       144867 non-null  object
 5   source_center                   144867 non-null  object
 6   source_name                     144574 non-null  object
 7   destination_center              144867 non-null  object
 8   destination_name                144606 non-null  object
 9   od_start_time                   144867 non-null  datetime64[ns]
 10  od_end_time                     144867 non-null  datetime64[ns]
 11  start_scan_to_end_scan          144867 non-null  float64
 12  is_cutoff                       144867 non-null  bool
 13  cutoff_factor                   144867 non-null  int64
 14  cutoff_timestamp                144867 non-null  object
 15  actual_distance_to_destination  144867 non-null  float64
 16  actual_time                     144867 non-null  float64
 17  osrm_time                       144867 non-null  float64
 18  osrm_distance                   144867 non-null  float64
 19  factor                          144867 non-null  float64
 20  segment_actual_time             144867 non-null  float64
 21  segment_osrm_time               144867 non-null  float64
 22  segment_osrm_distance           144867 non-null  float64
 23  segment_factor                  144867 non-null  float64
dtypes: bool(1), datetime64[ns](3), float64(10), int64(1), object(9)
memory usage: 25.6+ MB
```

In [19]:
```
for col in df.columns:
    print(col,':',df[col].nunique())
```

```
data : 2
trip_creation_time : 14817
route_schedule_uuid : 1504
route_type : 2
trip_uuid : 14817
source_center : 1508
source_name : 1498
destination_center : 1481
destination_name : 1468
od_start_time : 26369
od_end_time : 26369
start_scan_to_end_scan : 1915
is_cutoff : 2
cutoff_factor : 501
cutoff_timestamp : 93180
actual_distance_to_destination : 144515
actual_time : 3182
osrm_time : 1531
osrm_distance : 138046
factor : 45641
segment_actual_time : 747
segment_osrm_time : 214
segment_osrm_distance : 113799
segment_factor : 5675
```

In [20]:
```
df.isnull().sum()
```

Out[20]:

	0
data	0
trip_creation_time	0
route_schedule_uuid	0
route_type	0
trip_uuid	0
source_center	0
source_name	293
destination_center	0
destination_name	261
od_start_time	0
od_end_time	0
start_scan_to_end_scan	0
is_cutoff	0
cutoff_factor	0
cutoff_timestamp	0
actual_distance_to_destination	0
actual_time	0
osrm_time	0
osrm_distance	0
factor	0
segment_actual_time	0
segment_osrm_time	0

	0
segment_osrm_distance	0
segment_factor	0

dtype: int64

In this whole dataset only source and destination have missing values

In [21]:
```
def missing_to_df(data):
    total_missing_df = data.isnull().sum().sort_values(ascending =False)
    percent_missing_df = (data.isnull().sum()/len(data)*100).sort_values(ascending=False)
    missing_data_df = pd.concat([total_missing_df, percent_missing_df], axis=1, keys=['Total', 'Percent'])
    return missing_data_df
```

In [22]:
```
missing_df = missing_to_df(df)
missing_df[missing_df['Total'] > 0]
```

Out[22]:

	Total	Percent
source_name	293	0.202254
destination_name	261	0.180165

From source_name 0.20% data values are missing

From destination_name 0.18 % data values are missing

In [23]:
```
cat_missing = ['source_name', 'destination_name']
freq_imputer = SimpleImputer(strategy = 'most_frequent')
for col in cat_missing:
    df[col] = pd.DataFrame(freq_imputer.fit_transform(pd.DataFrame(df[col])))
```

In [24]:
```
missing_to_df(df)
```

Out[24]:

	Total	Percent
data	0	0.0
trip_creation_time	0	0.0
segment_osrm_distance	0	0.0
segment_osrm_time	0	0.0
segment_actual_time	0	0.0
factor	0	0.0
osrm_distance	0	0.0
osrm_time	0	0.0
actual_time	0	0.0
actual_distance_to_destination	0	0.0
cutoff_timestamp	0	0.0
cutoff_factor	0	0.0
is_cutoff	0	0.0
start_scan_to_end_scan	0	0.0
od_end_time	0	0.0
od_start_time	0	0.0
destination_name	0	0.0
destination_center	0	0.0
source_name	0	0.0
source_center	0	0.0
trip_uuid	0	0.0
route_type	0	0.0

	Total	Percent
route_schedule_uuid	0	0.0
segment_factor	0	0.0

In [25]:
```
df1 = df
```

In [26]:
```
df1_merge=df1.groupby(['trip_uuid','source_center','destination_center']).agg({'data':'first',
                                                                                'trip_creation_time':'first',
                                                                               'route_schedule_uuid':'first',
                                                                                'route_type':'first',
                                                                                'source_name':'first',
                                                                                'destination_name':'last',
                                                                                'od_start_time':'first',
                                                                                'od_end_time':'last',
                                                                                'start_scan_to_end_scan':'max',
                                                                                'actual_distance_to_destination':'max',
                                                                                'actual_time':'max',
                                                                                'osrm_time':'max',
                                                                                'osrm_distance':'max',
                                                                                'segment_actual_time':'sum',
                                                                                'segment_osrm_time':'sum',
                                                                                'segment_osrm_distance':'sum'}).reset_ir

df1_merge
```

Out[26]:

	trip_uuid	source_center	destination_center	data	trip_creation_time	route_schedule_uuid	route_type
0	trip-153671041653548748	IND209304AAA	IND000000ACB	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL
1	trip-153671041653548748	IND462022AAA	IND209304AAA	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL
2	trip-153671042288605164	IND561203AAB	IND562101AAA	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting
3	trip-153671042288605164	IND572101AAA	IND561203AAB	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting
4	trip-153671043369099517	IND000000ACB	IND160002AAC	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL
...	...	...	...	...	...	...	...
26363	trip-153861115439069069	IND628204AAA	IND627657AAA	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting
26364	trip-153861115439069069	IND628613AAA	IND627005AAA	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting
26365	trip-153861115439069069	IND628801AAA	IND628204AAA	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting
26366	trip-153861118270144424	IND583119AAA	IND583101AAA	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL
26367	trip-153861118270144424	IND583201AAA	IND583119AAA	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL

26368 rows × 19 columns

In [27]:
```
df1=df1_merge.groupby('trip_uuid').agg({'data':'first',
                                        'trip_creation_time':'first',
                                         'route_schedule_uuid':'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':'sum',
                                        'actual_distance_to_destination':'sum',
                                        'actual_time':'sum',
                                        'osrm_time':'sum',
                                        'osrm_distance':'sum',
                                        'segment_actual_time':'sum',
                                        'segment_osrm_time':'sum',
                                        'segment_osrm_distance':'sum'}).reset_index()
df1
```

Out[27]:

	trip_uuid	data	trip_creation_time	route_schedule_uuid	route_type	source_center	source_n
0	trip-153671041653548748	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	IND209304AAA	Kanpur_Central (Uttar Prad
1	trip-153671042288605164	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	IND561203AAB	Doddablpur_ChikaD (Karna
2	trip-153671043369099517	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	IND000000ACB	Gurgaon_Bilaspu (Hary
3	trip-153671046011330457	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	IND400072AAB	Mumbai (Maharas
4	trip-153671052974046625	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	IND583101AAA	Bellary_Dc (Karna
...	...	...	...	...	...	...	
14812	trip-153861095625827784	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	IND160002AAC	Chandigarh_Mehmdp (Pu
14813	trip-153861104386292051	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	IND121004AAB	FBD_Balabhgarh (Hary
14814	trip-153861106442901555	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	IND208006AAA	Kanpur_GovndNg (Uttar Prad
14815	trip-153861115439069069	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	IND627005AAA	Tirunelveli_Vdkku (Tamil N
14816	trip-153861118270144424	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	IND583119AAA	Sandur_WrdN1D (Karna

14817 rows × 19 columns

Extrqacting some features for actual analysis

In [28]:
```
df1[['destination','dest_state']] = df1['destination_name'].str.split('(', n=1, expand=True)
df1['dest_state'] = df1['dest_state'].str.rstrip(')')
df1[['dest_City','dest_place','dest_code']] = df1['destination'].str.split('_', n=2, expand=True)
df1[['source','source_state']] = df1['source_name'].str.split('(', n=1, expand=True)
df1['source_state'] = df1['source_state'].str.rstrip(')')
df1[['source_City','source_place','source_code']] = df1['source'].str.split('_', n=2, expand=True)
```

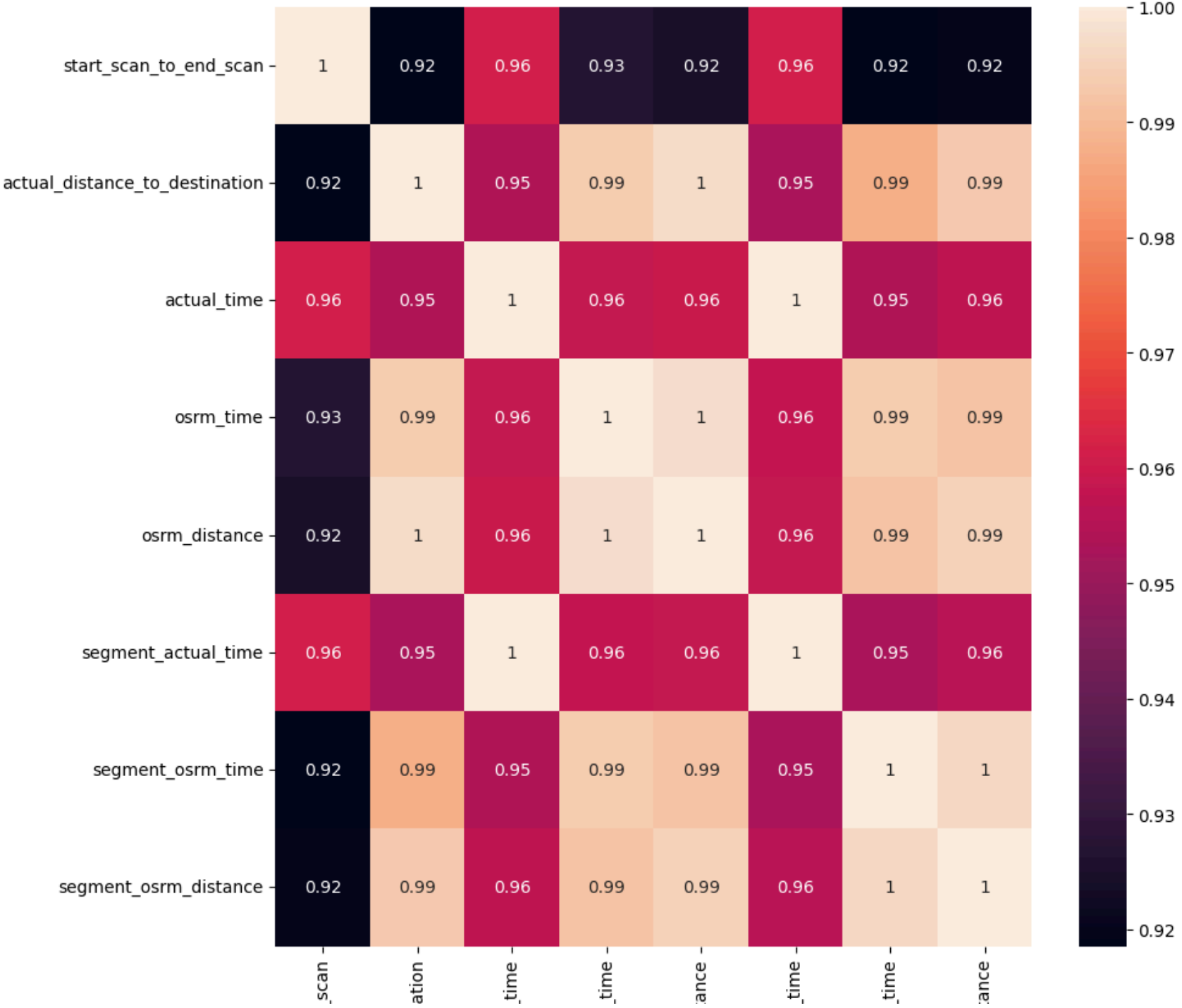
In [29]:
```
df1['trip_creation_year'] =df1['trip_creation_time'].dt.year
df1['trip_creation_month'] =df1['trip_creation_time'].dt.month_name()
df1['trip_creation_day'] =df1['trip_creation_time'].dt.day
df1
```

Out[29]:

	trip_uuid	data	trip_creation_time	route_schedule_uuid	route_type	source_center	source_n
0	trip-153671041653548748	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	IND209304AAA	Kanpur_Central (Uttar Prad
1	trip-153671042288605164	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	IND561203AAB	Doddablpur_ChikaD (Karna
2	trip-153671043369099517	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	IND000000ACB	Gurgaon_Bilaspu (Hary
3	trip-153671046011330457	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	IND400072AAB	Mumbai (Maharas
4	trip-153671052974046625	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	IND583101AAA	Bellary_Dc (Karna
...	...	...	...	...	...	...	
14812	trip-153861095625827784	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	IND160002AAC	Chandigarh_Mehmdp (Pu
14813	trip-153861104386292051	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	IND121004AAB	FBD_Balabhgarh (Hary
14814	trip-153861106442901555	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	IND208006AAA	Kanpur_GovndNg (Uttar Prad
14815	trip-153861115439069069	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	IND627005AAA	Tirunelveli_Vdkku (Tamil N
14816	trip-153861118270144424	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	IND583119AAA	Sandur_WrdN1D (Karna

14817 rows × 32 columns

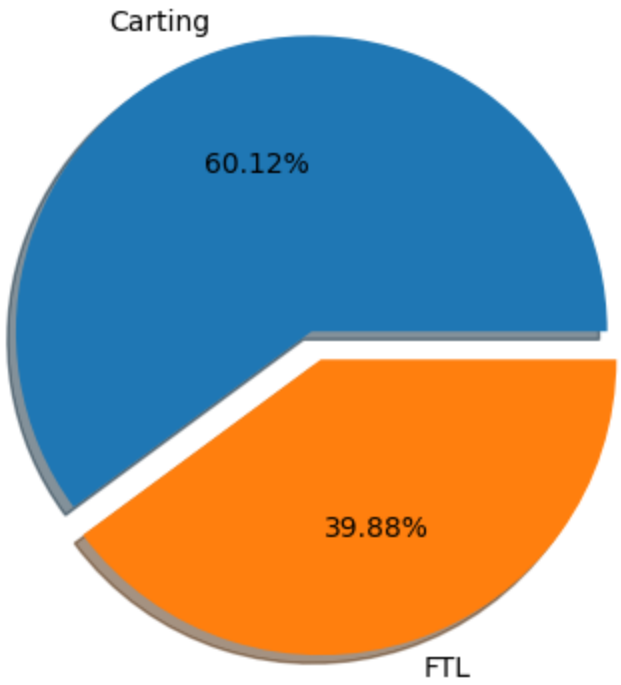
In [30]:
```
numeric_cols = df1.select_dtypes(include=['float64', 'int64']).columns
correlation_matrix = df1[numeric_cols].corr()

# Create heatmap
plt.figure(figsize=(10, 10))
sns.heatmap(correlation_matrix, annot=True)
plt.show()
```


start_scan_to_end_scan
actual_distance_to_destination
actual_time
osrm_time
osrm_distance
segment_actual_time
segment_osrm_time
segment_osrm_distance
1 0.92 0.96 0.93 0.92 0.96 0.92 0.92
0.92 1 0.95 0.99 1 0.95 0.99 0.99
0.96 0.95 1 0.96 0.96 1 0.95 0.96
0.93 0.99 0.96 1 1 0.96 0.99 0.99
0.92 1 0.96 1 1 0.96 0.99 0.99
0.96 0.95 1 0.96 0.96 1 0.95 0.96
0.92 0.99 0.95 0.99 0.99 0.95 1 1
0.92 0.99 0.96 0.99 0.99 0.96 1 1
scan
ation
time
time
ance
time
time
ance
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92

start_scan_to_end_ actual_distance_to_destin actual_ osrm_ osrm_dist segment_actual_ segment_osrm_ segment_osrm_dist

most prefered route type for delivery

In [31]:
```
plt.pie(data=df1,x=df1['route_type'].value_counts(),shadow=True,labels=['Carting','FTL'],explode=(0,0.1),autopct='%0
plt.plot()
plt.show()
```



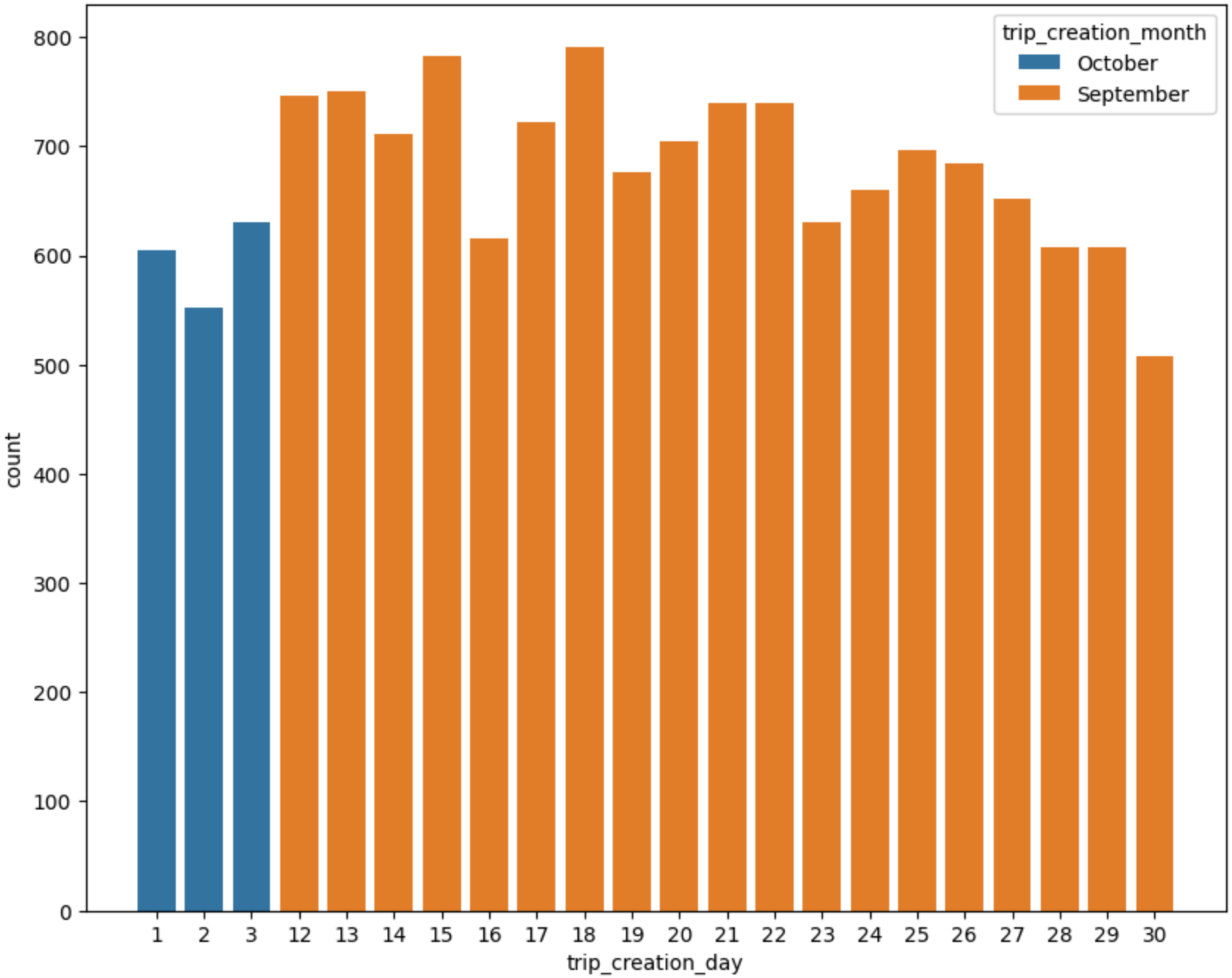
Most Prefered route type is carting

- Carting - 60.12%
- FTL-39.88%

Trip Creation day

In [32]:
```
plt.figure(figsize=(10,8))
sns.countplot(x=df1['trip_creation_day'],hue=df1['trip_creation_month'])
plt.plot()
```

Out[32]:
```
[]
```

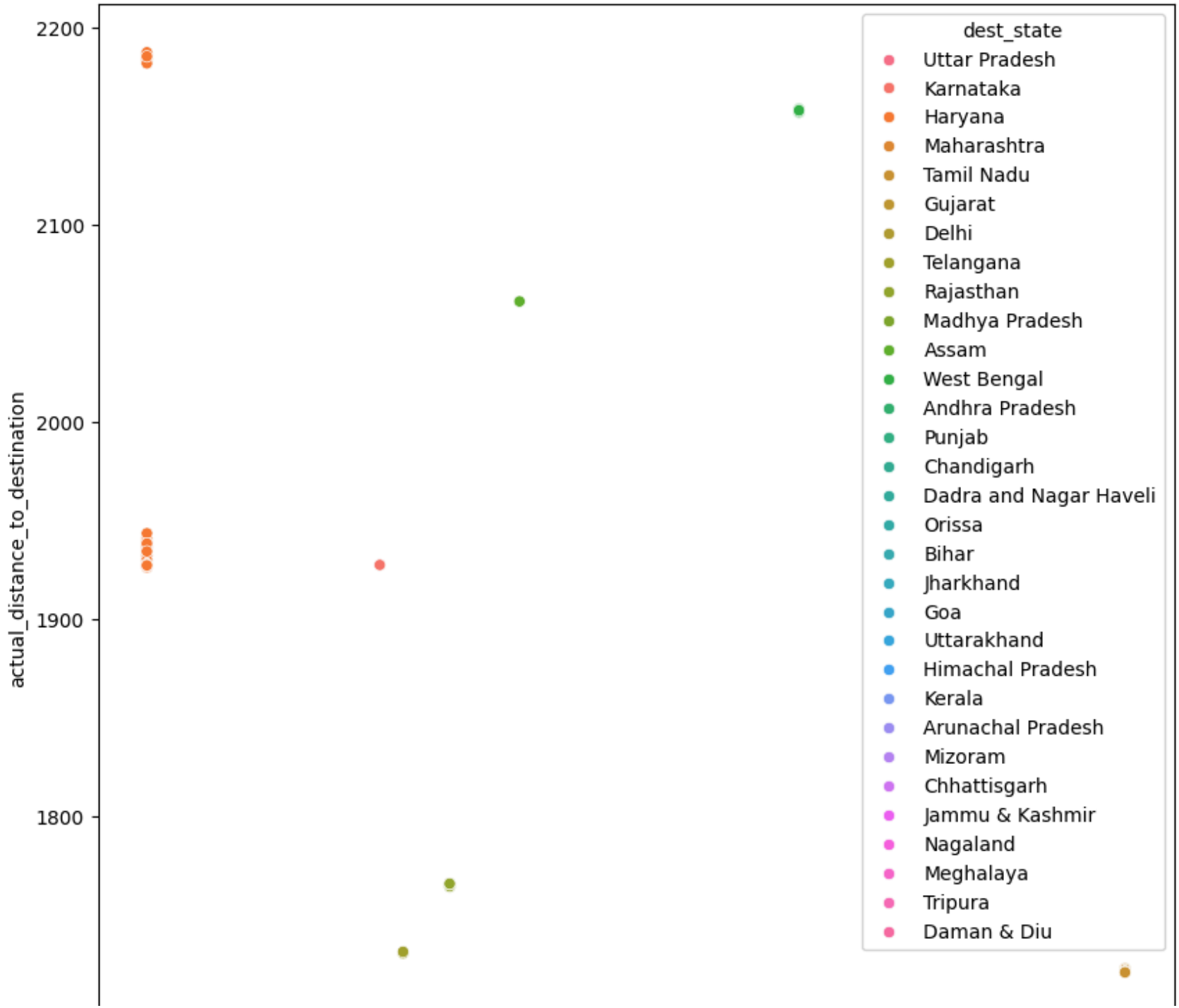

trip_creation_month
October
September
count
trip_creation_day
0
100
200
300
400
500
600
700
800
1
2
3
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Maximum trip is in september that to after 12th till the end of the month and only three consecutive trips are noted in october on starting of month

Actual distance to destination from source city

In [33]:
```
y=df1['actual_distance_to_destination'].sort_values(ascending=False).head(100).to_frame()
y
plt.figure(figsize=(10,10))
sns.scatterplot(x=df1['dest_City'],y=y['actual_distance_to_destination'],hue=df1['dest_state'])
plt.xticks(rotation=90)
plt.plot()
```

Out[33]:
```
[]
```


actual_distance_to_destination
2200
2100
2000
1900
1800
dest_state
Uttar Pradesh
Karnataka
Haryana
Maharashtra
Tamil Nadu
Gujarat
Delhi
Telangana
Rajasthan
Madhya Pradesh
Assam
West Bengal
Andhra Pradesh
Punjab
Chandigarh
Dadra and Nagar Haveli
Orissa
Bihar
Jharkhand
Goa
Uttarakhand
Himachal Pradesh
Kerala
Arunachal Pradesh
Mizoram
Chhattisgarh
Jammu & Kashmir
Nagaland
Meghalaya
Tripura
Daman & Diu

Indian states with source and destination delivery count

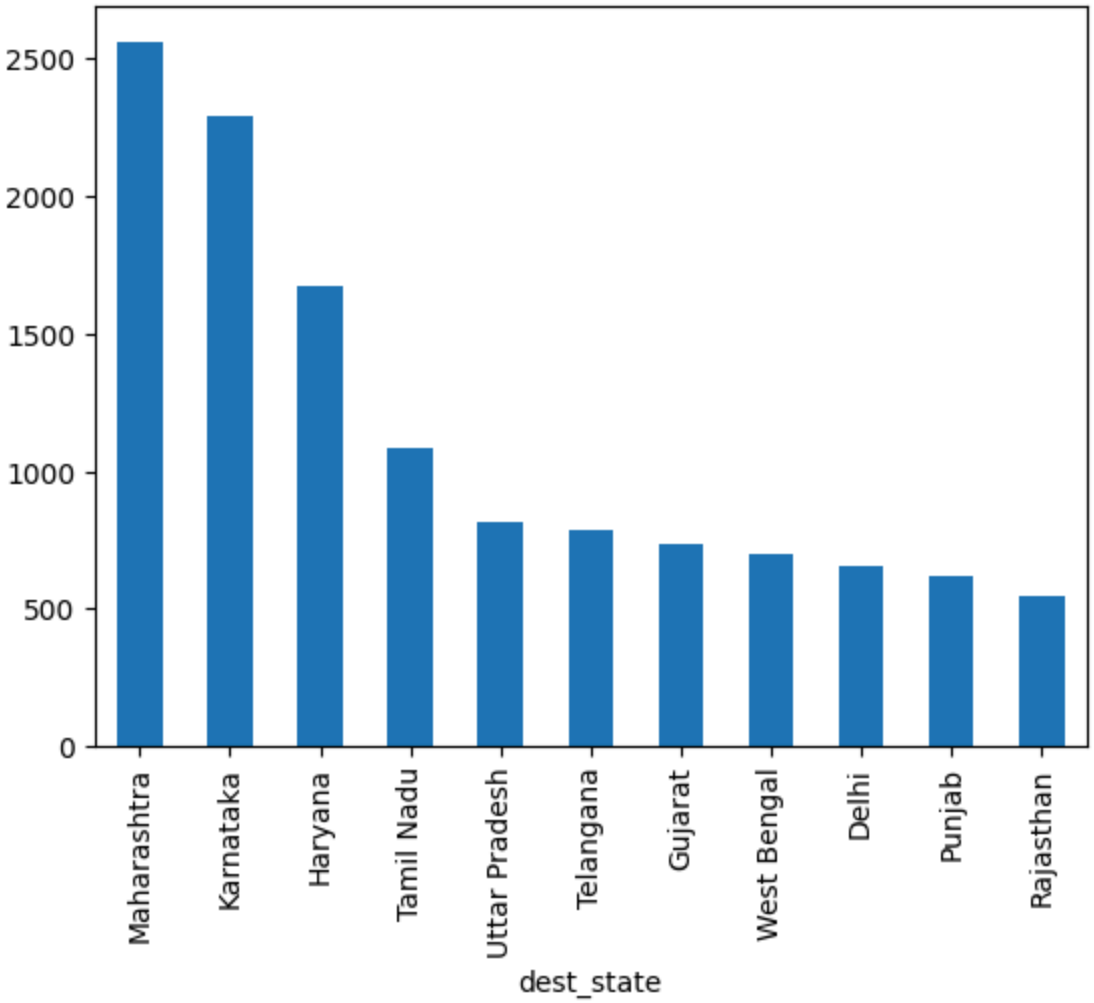
In [34]:
```
x=df1.groupby('dest_state')['trip_uuid'].count().sort_values(ascending=False).head(11).to_frame().reset_index()
x
```

Out[34]:

	dest_state	trip_uuid
0	Maharashtra	2561
1	Karnataka	2294
2	Haryana	1670
3	Tamil Nadu	1084
4	Uttar Pradesh	811
5	Telangana	784
6	Gujarat	734
7	West Bengal	697
8	Delhi	652
9	Punjab	617
10	Rajasthan	543

In [35]:
```
df1.groupby('dest_state')['trip_uuid'].count().sort_values(ascending=False).head(11).plot(kind='bar')
```

Out[35]: <Axes: xlabel='dest_state'>



Highest delivery is done in Maharastra:2561 and Second Highest is Karnataka: Top 5 States with higher deliveries are

- Maharashtra

- Karnataka
- Haryana
- Tamilnadu
- UttarPradesh

Source state with delivery counts

In [36]:
```
x=df1.groupby('source_state')['trip_uuid'].count().sort_values(ascending=False).head(11).to_frame().reset_index()
x
```

Out[36]:

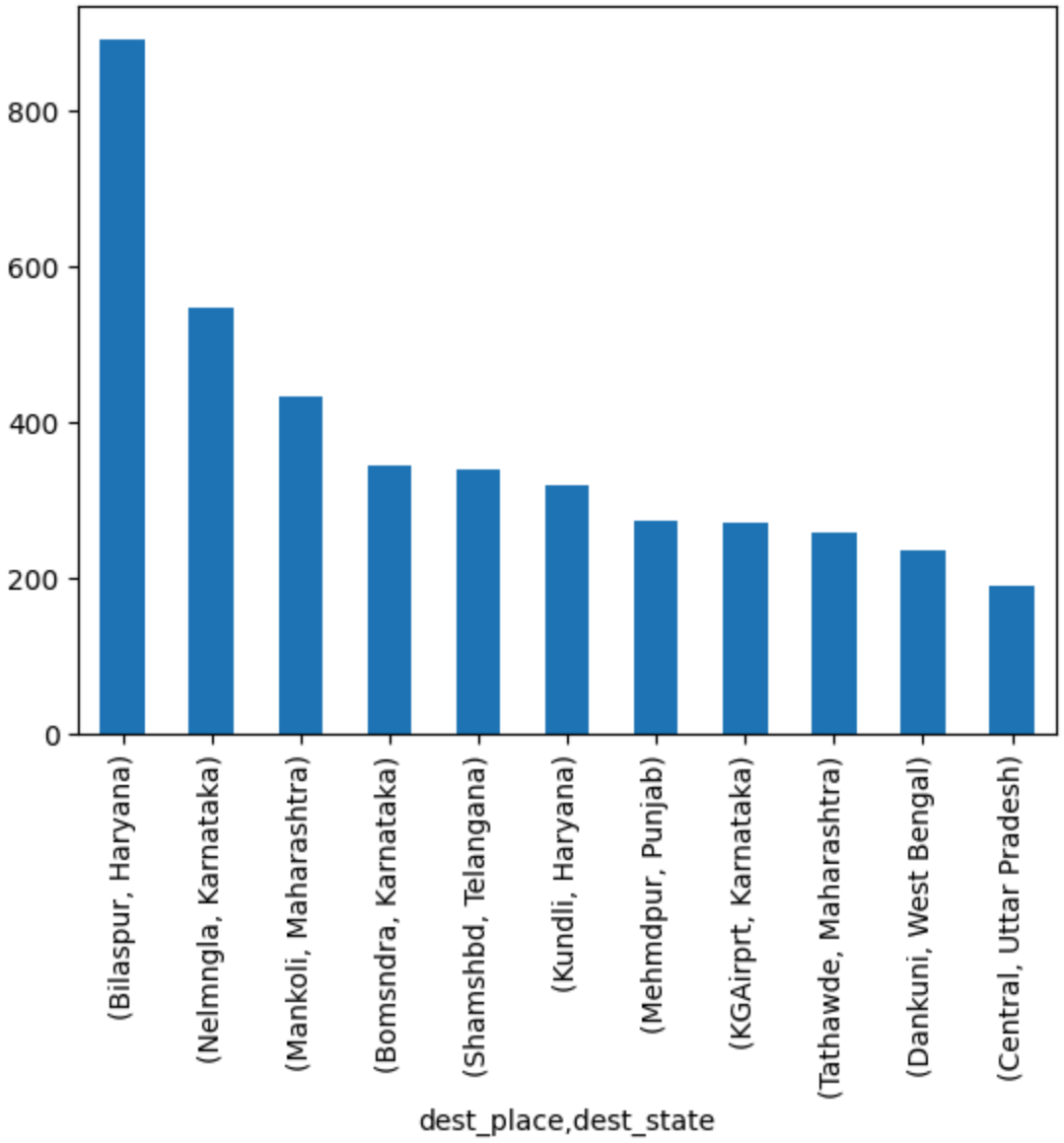
	source_state	trip_uuid
0	Maharashtra	2714
1	Karnataka	2143
2	Haryana	1854
3	Tamil Nadu	1039
4	Telangana	781
5	Uttar Pradesh	762
6	Gujarat	750
7	Delhi	728
8	West Bengal	665
9	Punjab	536
10	Rajasthan	514

The sourcestate Maharashtra have delivery around 2714 After that karnataka : 2143

Top 5 Indian states with most product delivery sources are : Maharashtra Karnataka Haryana Tamil Nadu and Telangana

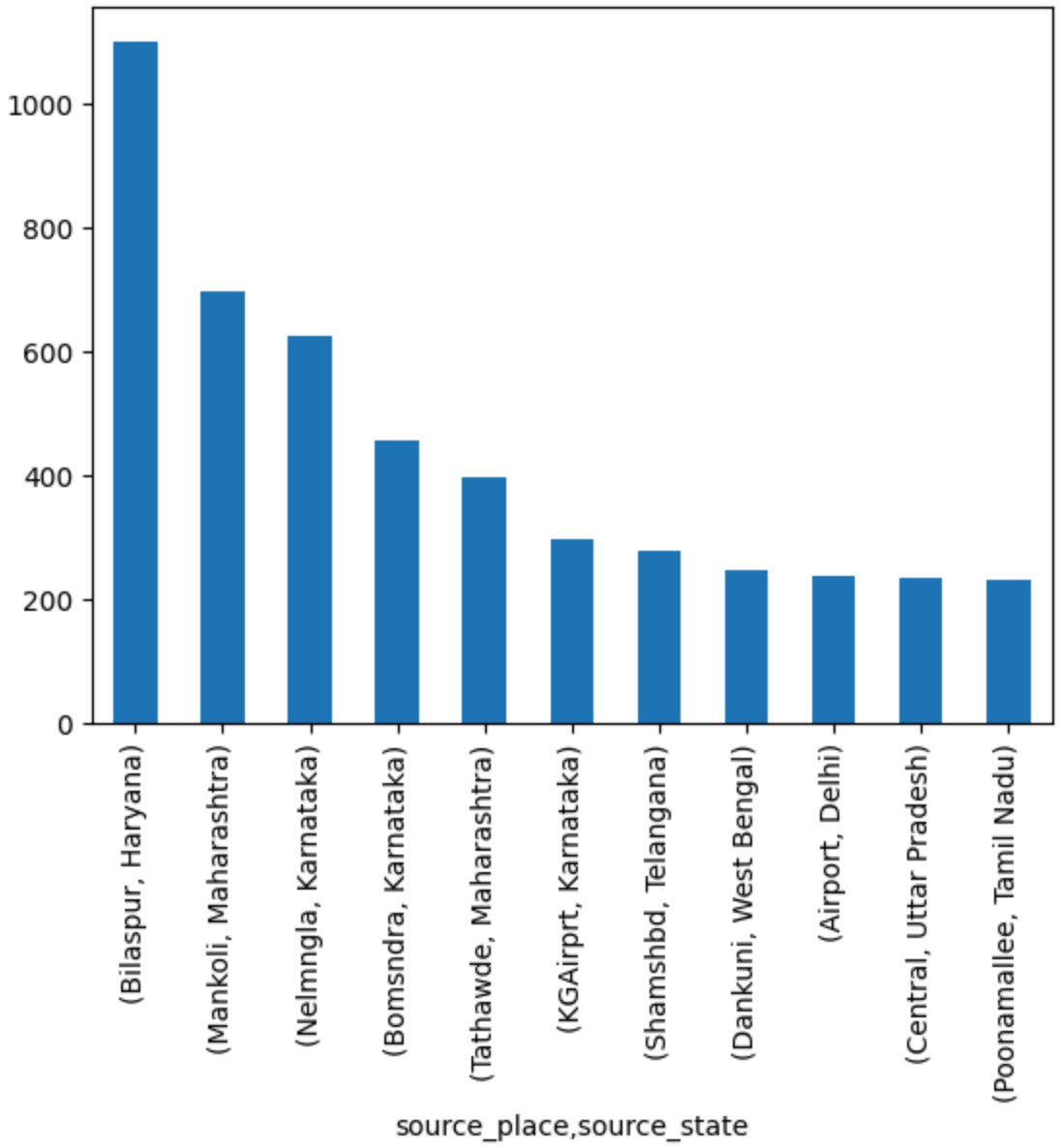
In [37]:
```
df1.groupby(['dest_place','dest_state'])['trip_uuid'].count().sort_values(ascending=False).head(11).plot(kind='bar')
```

Out[37]: <Axes: xlabel='dest_place,dest_state'>



In [38]:
```
df1.groupby(['source_place','source_state'])['trip_uuid'].count().sort_values(ascending=False).head(11).plot(kind='ba
```

Out[38]: <Axes: xlabel='source_place,source_state'>



In [39]:
```
h=df1.groupby(['dest_City','dest_state'])['trip_uuid'].count().sort_values(ascending=False).head(11).to_frame().reset
h
```

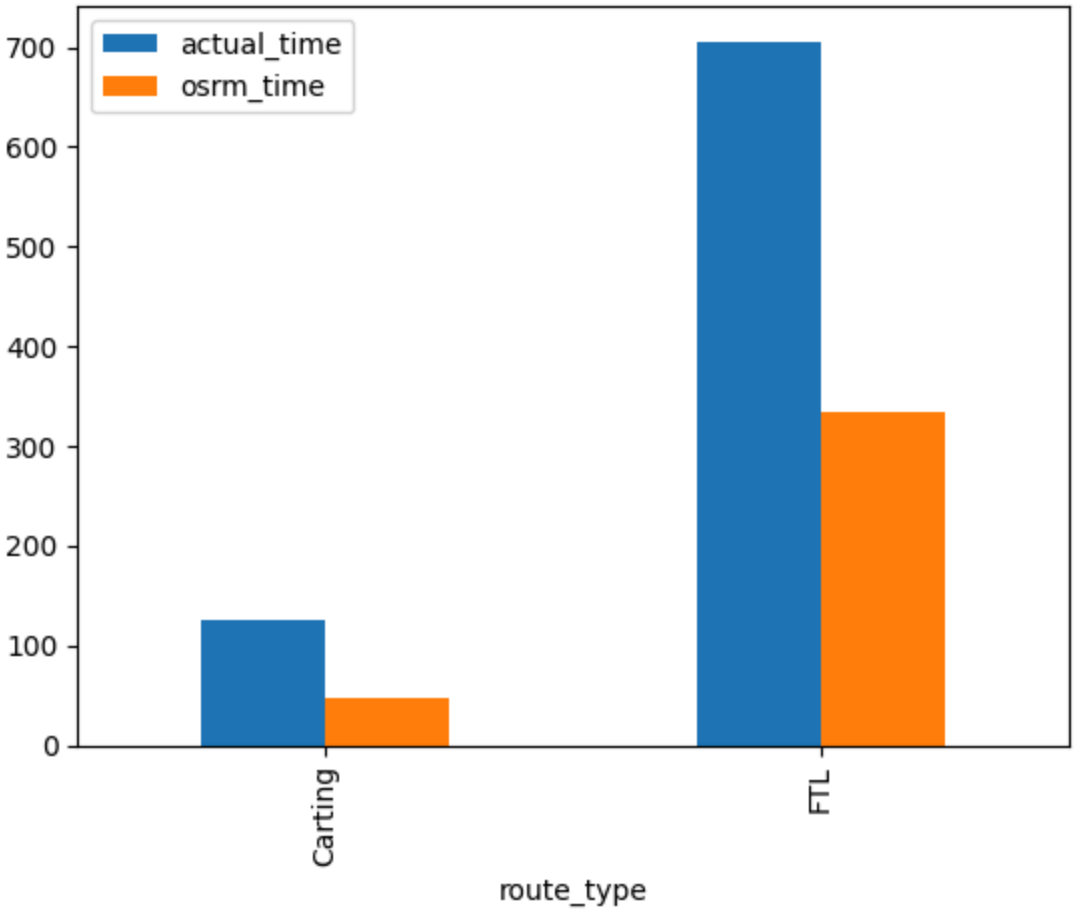
Out[39]:

	dest_City	dest_state	trip_uuid
0	Bengaluru	Karnataka	1088
1	Mumbai	Maharashtra	966
2	Gurgaon	Haryana	904
3	Bangalore	Karnataka	551
4	Delhi	Delhi	549
5	Hyderabad	Telangana	499
6	Bhiwandi	Maharashtra	434
7	Chennai	Tamil Nadu	410
8	Sonipat	Haryana	322
9	Pune	Maharashtra	313
10	Kolkata	West Bengal	277

By analysing through each city Bengaluru in Karnataka have more delivery compaired to other cities The cities with most destination centres are : Bangalore Mumbai Gurgaon Bangalore and Delhi

Which route_type take more time to reach destination

In [40]:
```
df1.groupby('route_type').aggregate({'actual_time':'mean','osrm_time':'mean'}).plot(kind='bar')
plt.show()
```



We can say that FTL shipment takes more time for delivery

In [41]:
```
df1['weekday'] = df1['trip_creation_time'].dt.day_name()
df1
```

Out[41]:

	trip_uuid	data	trip_creation_time	route_schedule_uuid	route_type	source_center	source_n
0	trip-153671041653548748	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	IND209304AAA	Kanpur_Central (Uttar Prad
1	trip-153671042288605164	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	IND561203AAB	Doddablpur_ChikaD (Karna
2	trip-153671043369099517	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	IND000000ACB	Gurgaon_Bilaspu (Hary
3	trip-153671046011330457	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	IND400072AAB	Mumbai (Maharas
4	trip-153671052974046625	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	IND583101AAA	Bellary_Dc (Karna
...	...	...	...	...	...	...	
14812	trip-153861095625827784	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	IND160002AAC	Chandigarh_Mehmdp (Pu
14813	trip-153861104386292051	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	IND121004AAB	FBD_Balabhgarh (Hary
14814	trip-153861106442901555	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	IND208006AAA	Kanpur_GovndNg (Uttar Prad
14815	trip-153861115439069069	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	IND627005AAA	Tirunelveli_Vdkku (Tamil N
14816	trip-153861118270144424	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	IND583119AAA	Sandur_WrdN1D (Karna

14817 rows × 33 columns

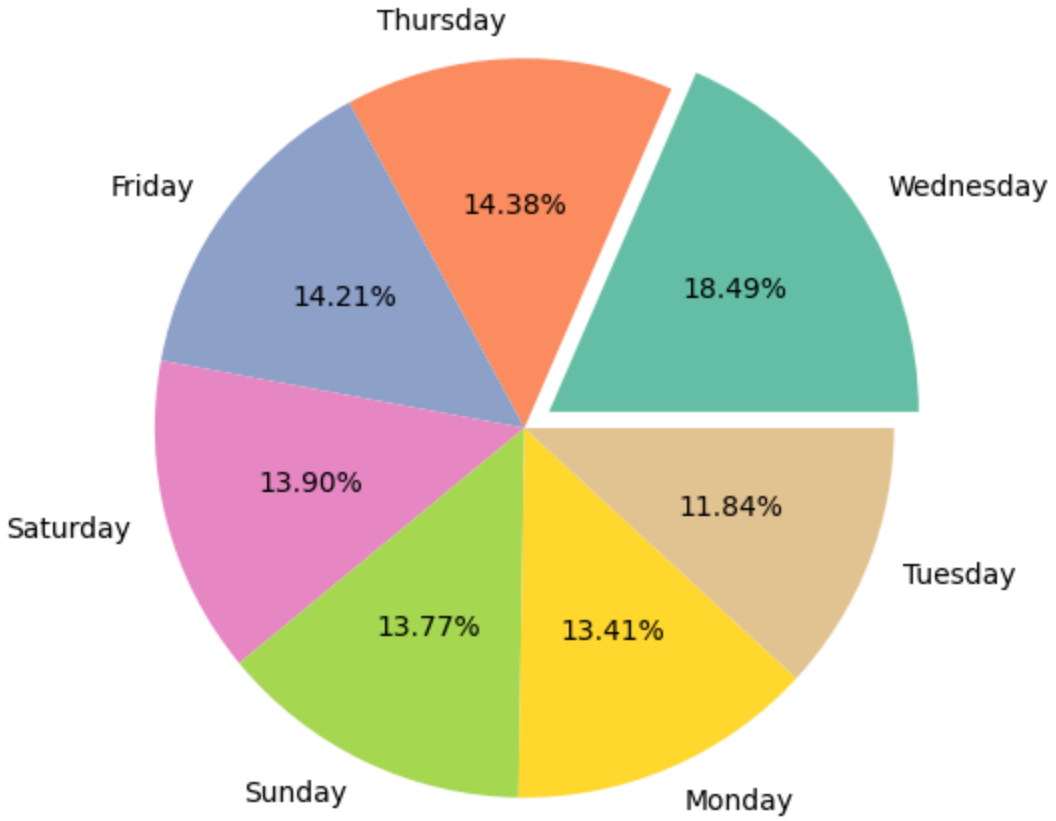
In [42]:
```
x=df1['weekday'].value_counts().to_frame('count').reset_index()
x
```

Out[42]:

	weekday	count
0	Wednesday	2739
1	Saturday	2130
2	Thursday	2106
3	Friday	2060
4	Tuesday	2040
5	Monday	1987
6	Sunday	1755

In [43]:
```
plt.figure(figsize=(8,6))
palette_color = sns.color_palette('Set2')
plt.pie(data=x, x=x['count'], colors=palette_color, labels=['Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday',
plt.plot()
```

Out[43]:
```
[]
```



In [44]:
```
x1=df1.groupby('source_state').agg({'actual_time':'mean','osrm_time':'mean'}).reset_index()
x1=pd.melt(x1, id_vars=['source_state'], value_vars=['actual_time', 'osrm_time'])
x1
```

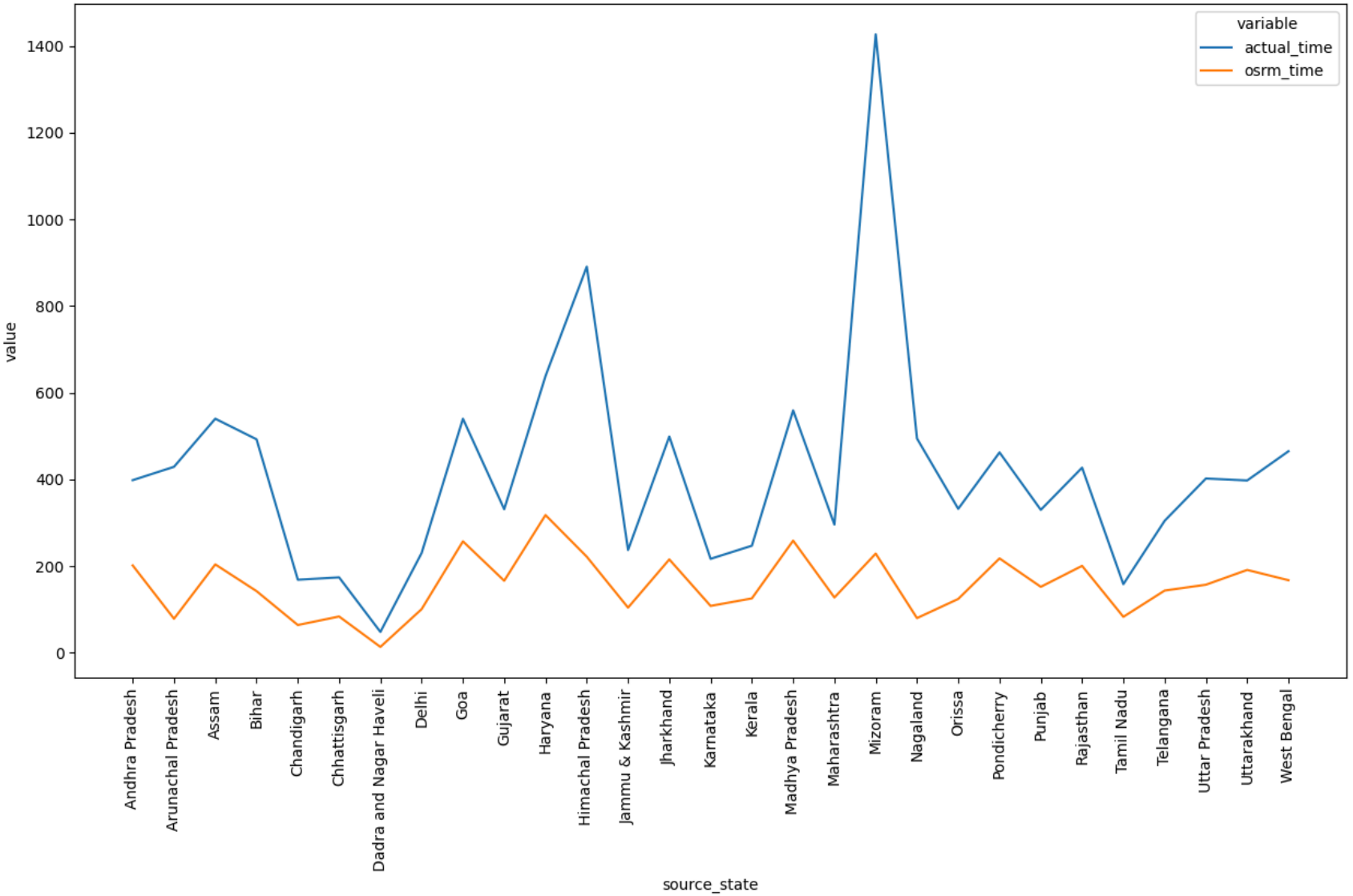
Out[44]:

	source_state	variable	value
0	Andhra Pradesh	actual_time	398.435484
1	Arunachal Pradesh	actual_time	429.250000
2	Assam	actual_time	540.171642
3	Bihar	actual_time	492.645714
4	Chandigarh	actual_time	168.741935
5	Chhattisgarh	actual_time	174.139535
6	Dadra and Nagar Haveli	actual_time	48.333333
7	Delhi	actual_time	230.550824
8	Goa	actual_time	540.138462
9	Gujarat	actual_time	331.470667
10	Haryana	actual_time	638.527508
11	Himachal Pradesh	actual_time	891.088235
12	Jammu & Kashmir	actual_time	237.235294
13	Jharkhand	actual_time	498.981250
14	Karnataka	actual_time	216.933738
15	Kerala	actual_time	247.173010
16	Madhya Pradesh	actual_time	559.406940
17	Maharashtra	actual_time	296.011054
18	Mizoram	actual_time	1427.000000
19	Nagaland	actual_time	494.600000
20	Orissa	actual_time	332.308411
21	Pondicherry	actual_time	462.416667

	source_state	variable	value
22	Punjab	actual_time	329.917910
23	Rajasthan	actual_time	427.264591
24	Tamil Nadu	actual_time	158.486044
25	Telangana	actual_time	304.624840
26	Uttar Pradesh	actual_time	402.257218
27	Uttarakhand	actual_time	397.728070
28	West Bengal	actual_time	465.124812
29	Andhra Pradesh	osrm_time	201.670507
30	Arunachal Pradesh	osrm_time	78.750000
31	Assam	osrm_time	204.085821
32	Bihar	osrm_time	142.431429
33	Chandigarh	osrm_time	64.215054
34	Chhattisgarh	osrm_time	83.906977
35	Dadra and Nagar Haveli	osrm_time	13.733333
36	Delhi	osrm_time	100.934066
37	Goa	osrm_time	257.276923
38	Gujarat	osrm_time	166.368000
39	Haryana	osrm_time	317.956311
40	Himachal Pradesh	osrm_time	221.852941
41	Jammu & Kashmir	osrm_time	104.352941
42	Jharkhand	osrm_time	215.600000
43	Karnataka	osrm_time	108.295847

	source_state	variable	value
44	Kerala	osrm_time	125.650519
45	Madhya Pradesh	osrm_time	258.952681
46	Maharashtra	osrm_time	127.739499
47	Mizoram	osrm_time	229.000000
48	Nagaland	osrm_time	80.200000
49	Orissa	osrm_time	124.411215
50	Pondicherry	osrm_time	217.916667
51	Punjab	osrm_time	152.354478
52	Rajasthan	osrm_time	200.708171
53	Tamil Nadu	osrm_time	83.142445
54	Telangana	osrm_time	143.732394
55	Uttar Pradesh	osrm_time	157.217848
56	Uttarakhand	osrm_time	191.324561
57	West Bengal	osrm_time	167.622556

In [45]:
```
plt.figure(figsize=(15,8))
sns.lineplot(data=x1,x='source_state',y='value',hue='variable')
plt.xticks(rotation=90)
plt.show()
```



Feature engineering part

In [46]:
```
df1['total_min_diff']=(df1['od_end_time']-df1['od_start_time'])/pd.Timedelta(minutes=1)
df1
```

Out[46]:

	trip_uuid	data	trip_creation_time	route_schedule_uuid	route_type	source_center	source_n
0	trip-153671041653548748	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	IND209304AAA	Kanpur_Central (Uttar Prad
1	trip-153671042288605164	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	IND561203AAB	Doddablpur_ChikaD (Karna
2	trip-153671043369099517	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	IND000000ACB	Gurgaon_Bilaspu (Hary
3	trip-153671046011330457	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	IND400072AAB	Mumbai (Maharas
4	trip-153671052974046625	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	IND583101AAA	Bellary_Dc (Karna
...	...	...	...	...	...	...	
14812	trip-153861095625827784	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	IND160002AAC	Chandigarh_Mehmdp (Pu
14813	trip-153861104386292051	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	IND121004AAB	FBD_Balabhgarh (Hary
14814	trip-153861106442901555	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	IND208006AAA	Kanpur_GovndNg (Uttar Prad
14815	trip-153861115439069069	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	IND627005AAA	Tirunelveli_Vdkku (Tamil N
14816	trip-153861118270144424	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	IND583119AAA	Sandur_WrdN1D (Karna

14817 rows × 34 columns

Compare the difference between total_min_diff and start_scan_to_end_scan. Do hypothesis testing/ Visual analysis to check.

In [ ]:

In [47]:
```
df1[['start_scan_to_end_scan','total_min_diff']]
```

Out[47]:

	start_scan_to_end_scan	total_min_diff
0	2259.0	0.000000
1	180.0	0.000000
2	3933.0	0.000000
3	100.0	100.494935
4	717.0	232.556228
...	...	...
14812	257.0	405.485842
14813	60.0	60.590521
14814	421.0	0.000000
14815	347.0	149.831354
14816	353.0	0.000000

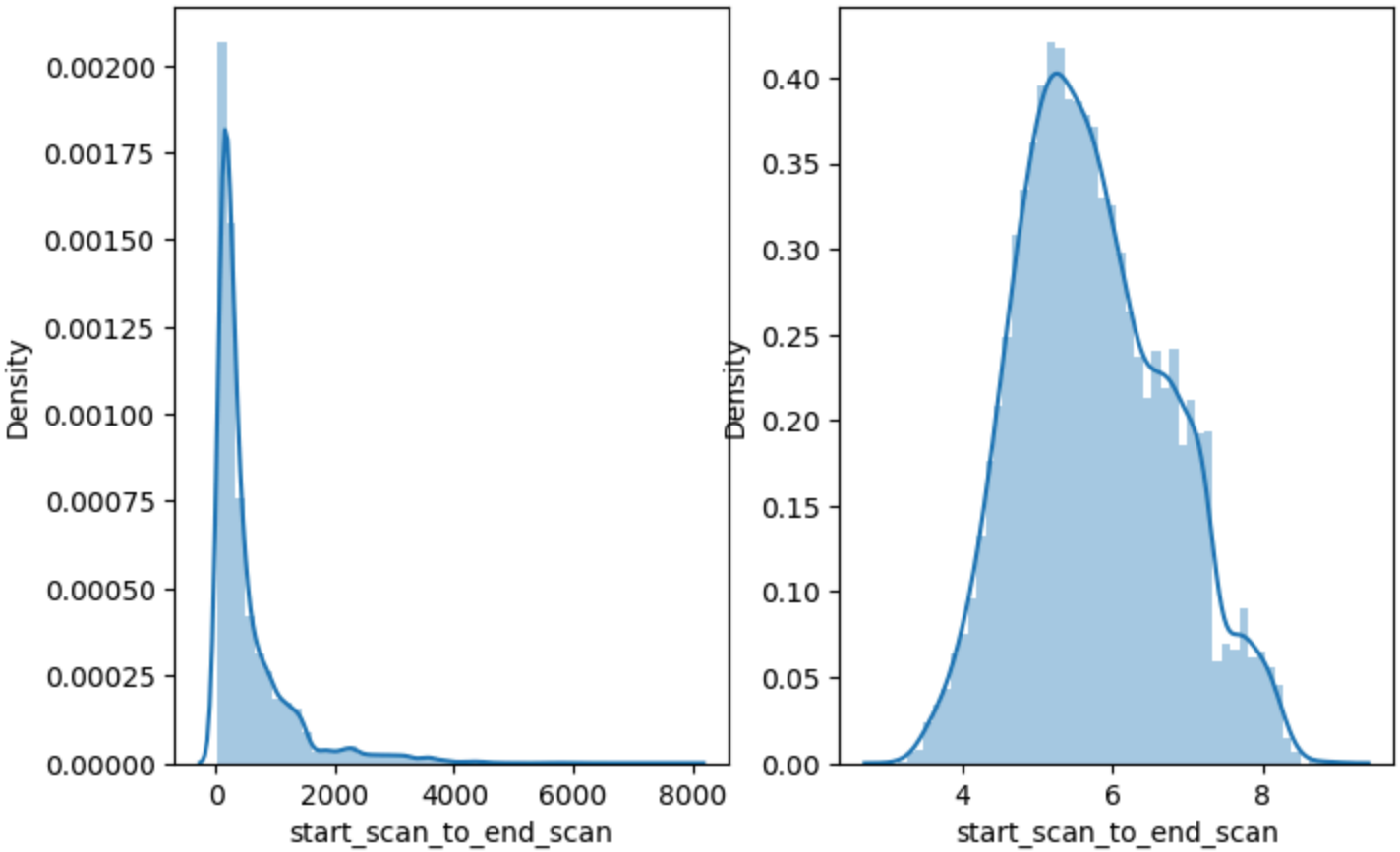
14817 rows × 2 columns

In [ ]:
```
#Hypothesis Test
#Ho= mean (start_scan_to_end_scan )==mean(total_min_diff)
#Ha=mean (start_scan_to_end_scan )!=mean(total_min_diff)
```

In [49]:
```
#Checking normal distribution of start_scan_to_end_scan
plt.figure(figsize=(8,5))
```

```
plt.subplot(121)
sns.distplot(df1['start_scan_to_end_scan'])
plt.plot()
plt.subplot(122)
sns.distplot(np.log(df1['start_scan_to_end_scan']))
plt.plot()
```

Out[49]: []



- Here we can see in first figure there is more outlier and it is lognormal distribution
- By Hypothesis we need to check that the mean value of observed column 'norm_total_min_diff' inside the region of 'norm_start_scan_to_end_scan' or not. By this we can accept or reject hypothesis statement Ho and Ha

In [50]:
```
x=df1[['start_scan_to_end_scan','total_min_diff']]
x
```

Out[50]:

	start_scan_to_end_scan	total_min_diff
0	2259.0	0.000000
1	180.0	0.000000
2	3933.0	0.000000
3	100.0	100.494935
4	717.0	232.556228
...	...	...
14812	257.0	405.485842
14813	60.0	60.590521
14814	421.0	0.000000
14815	347.0	149.831354
14816	353.0	0.000000

14817 rows × 2 columns

In [51]:
```
observed_mean_difference=np.mean(x['start_scan_to_end_scan'])-np.mean(x['total_min_diff'])
observed_mean_difference
```

Out[51]:
```
160.16246217165082
```

In [54]:
```
y=df1.groupby(['trip_uuid','source_center'])['actual_time'].max().to_frame('time').reset_index()
print(y)
y.groupby('trip_uuid')['time'].sum().to_frame('actual_time').reset_index()
```

```
                      trip_uuid source_center    time
0      trip-153671041653548748  IND209304AAA  1562.0
1      trip-153671042288605164  IND561203AAB   143.0
2      trip-153671043369099517  IND000000ACB  3347.0
3      trip-153671046011330457  IND400072AAB    59.0
4      trip-153671052974046625  IND583101AAA   341.0
...                        ...           ...     ...
14812  trip-153861095625827784  IND160002AAC    83.0
14813  trip-153861104386292051  IND121004AAB    21.0
14814  trip-153861106442901555  IND208006AAA   282.0
14815  trip-153861115439069069  IND627005AAA   264.0
14816  trip-153861118270144424  IND583119AAA   275.0

[14817 rows x 3 columns]
```

Out[54]:

	trip_uuid	actual_time
0	trip-153671041653548748	1562.0
1	trip-153671042288605164	143.0
2	trip-153671043369099517	3347.0
3	trip-153671046011330457	59.0
4	trip-153671052974046625	341.0
...	...	...
14812	trip-153861095625827784	83.0
14813	trip-153861104386292051	21.0
14814	trip-153861106442901555	282.0
14815	trip-153861115439069069	264.0
14816	trip-153861118270144424	275.0

14817 rows × 2 columns

In [55]:
```
z=df1.groupby(['trip_uuid','source_center'])['osrm_time'].max().to_frame('osr_times').reset_index()
c=z[z['trip_uuid']=='trip-153671041653548748']
```

```
c
```

Out[55]:

	trip_uuid	source_center	osr_times
0	trip-153671041653548748	IND209304AAA	743.0

In [56]:
```
z=df1.groupby(['trip_uuid','source_center'])['osrm_time'].max().to_frame('osr_times').reset_index()
z.groupby('trip_uuid')['osr_times'].sum().to_frame('actual_osr_time').reset_index()
```

Out[56]:

	trip_uuid	actual_osr_time
0	trip-153671041653548748	743.0
1	trip-153671042288605164	68.0
2	trip-153671043369099517	1741.0
3	trip-153671046011330457	15.0
4	trip-153671052974046625	117.0
...	...	...
14812	trip-153861095625827784	62.0
14813	trip-153861104386292051	12.0
14814	trip-153861106442901555	54.0
14815	trip-153861115439069069	184.0
14816	trip-153861118270144424	68.0

14817 rows × 2 columns

In [57]:
```
x=df1.groupby(['trip_uuid','source_center']).agg({'actual_time':'max','osrm_time':'max'})
x=x.groupby('trip_uuid').agg({'actual_time':'sum','osrm_time':'sum'}).reset_index()
x.rename(columns={'actual_time':'actual_time_agg','osrm_time':'osrm_time_agg'},inplace=True)
```

In [58]:
```
x
```

Out[58]:

	trip_uuid	actual_time_agg	osrm_time_agg
0	trip-153671041653548748	1562.0	743.0
1	trip-153671042288605164	143.0	68.0
2	trip-153671043369099517	3347.0	1741.0
3	trip-153671046011330457	59.0	15.0
4	trip-153671052974046625	341.0	117.0
...	...	...	...
14812	trip-153861095625827784	83.0	62.0
14813	trip-153861104386292051	21.0	12.0
14814	trip-153861106442901555	282.0	54.0
14815	trip-153861115439069069	264.0	184.0
14816	trip-153861118270144424	275.0	68.0

14817 rows × 3 columns

In [59]:
```
# mean difference of obsevation
mean_observed_diff=np.mean(x['actual_time_agg'])-np.mean(x['osrm_time_agg'])
mean_observed_diff
```

Out[59]:
```
195.07255179860968
```

- we can say that mean of observed_mean_difference outside region of the expected mean_test_sample or we can it is in rejection region
- It implies that mean of 'actual_time_agg' equal to 'osrm_time_agg'. So rejecting the null hypothesis

Do hypothesis testing/ visual analysis between actual_time aggregated value and segment actual time aggregated value (aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)

In [60]:
```
df1
```

Out[60]:

	trip_uuid	data	trip_creation_time	route_schedule_uuid	route_type	source_center	source_n
0	trip-153671041653548748	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	IND209304AAA	Kanpur_Central (Uttar Prad
1	trip-153671042288605164	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	IND561203AAB	Doddablpur_ChikaD (Karna
2	trip-153671043369099517	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	IND000000ACB	Gurgaon_Bilaspu (Hary
3	trip-153671046011330457	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	IND400072AAB	Mumbai (Maharas
4	trip-153671052974046625	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	IND583101AAA	Bellary_Dc (Karna
...	...	...	...	...	...	...	
14812	trip-153861095625827784	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	IND160002AAC	Chandigarh_Mehmdp (Pu
14813	trip-153861104386292051	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	IND121004AAB	FBD_Balabhgarh (Hary
14814	trip-153861106442901555	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	IND208006AAA	Kanpur_GovndNg (Uttar Prad
14815	trip-153861115439069069	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	IND627005AAA	Tirunelveli_Vdkku (Tamil N
14816	trip-153861118270144424	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	IND583119AAA	Sandur_WrdN1D (Karna

14817 rows × 34 columns

In [61]:
```
y=df1.groupby(['trip_uuid','source_center'])['segment_actual_time'].max().to_frame('segment_time').reset_index()
print(y)
y.groupby('trip_uuid')['segment_time'].sum().to_frame('segmented_actual_time_agg').reset_index()
```

```
                    trip_uuid source_center  segment_time
0      trip-153671041653548748  IND209304AAA        1548.0
1      trip-153671042288605164  IND561203AAB         141.0
2      trip-153671043369099517  IND000000ACB        3308.0
3      trip-153671046011330457  IND400072AAB          59.0
4      trip-153671052974046625  IND583101AAA         340.0
...                        ...           ...           ...
14812  trip-153861095625827784  IND160002AAC          82.0
14813  trip-153861104386292051  IND121004AAB          21.0
14814  trip-153861106442901555  IND208006AAA         281.0
14815  trip-153861115439069069  IND627005AAA         258.0
14816  trip-153861118270144424  IND583119AAA         274.0

[14817 rows x 3 columns]
```

Out[61]:

	trip_uuid	segmented_actual_time_agg
0	trip-153671041653548748	1548.0
1	trip-153671042288605164	141.0
2	trip-153671043369099517	3308.0
3	trip-153671046011330457	59.0
4	trip-153671052974046625	340.0
...	...	...
14812	trip-153861095625827784	82.0
14813	trip-153861104386292051	21.0
14814	trip-153861106442901555	281.0
14815	trip-153861115439069069	258.0
14816	trip-153861118270144424	274.0

14817 rows × 2 columns

In [62]:
```
w=df1.groupby(['trip_uuid','source_center']).agg({'actual_time':'max','segment_actual_time':'sum'})
w=w.groupby('trip_uuid').agg({'actual_time':'sum','segment_actual_time':'sum'}).reset_index()
w.rename(columns={'actual_time':'actual_time_agg','segment_actual_time':'segment_actual_time_agg'},inplace=True)
w
```

Out[62]:

	trip_uuid	actual_time_agg	segment_actual_time_agg
0	trip-153671041653548748	1562.0	1548.0
1	trip-153671042288605164	143.0	141.0
2	trip-153671043369099517	3347.0	3308.0
3	trip-153671046011330457	59.0	59.0
4	trip-153671052974046625	341.0	340.0
...	...	...	...
14812	trip-153861095625827784	83.0	82.0
14813	trip-153861104386292051	21.0	21.0
14814	trip-153861106442901555	282.0	281.0
14815	trip-153861115439069069	264.0	258.0
14816	trip-153861118270144424	275.0	274.0

14817 rows × 3 columns

In [63]:
```
#mean_difference_observation
mean_of_observation=np.mean(w['actual_time_agg'])-np.mean(w['segment_actual_time_agg'])
mean_of_observation
```

Out[63]:
```
3.251467908483505
```

In [64]:
```
# concate two observed columnss
s=np.concatenate((w['actual_time_agg'],w['segment_actual_time_agg']))
s
```

Out[64]:
```
array([1562.,  143., 3347., ...,  281.,  258.,  274.])
```

3.5 Do hypothesis testing/ visual analysis between 1 (aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)

In [65]:
```
w=df1.groupby(['trip_uuid','source_center']).agg({'osrm_distance':'max','segment_osrm_distance':'sum'})
w=w.groupby('trip_uuid').agg({'osrm_distance':'sum','segment_osrm_distance':'sum'}).reset_index()
w.rename(columns={'osrm_distance':'osrm_distance_agg','segment_osrm_distance':'segment_osrm_distance_agg'},inplace=Tr
w
```

Out[65]:

	trip_uuid	osrm_distance_agg	segment_osrm_distance_agg
0	trip-153671041653548748	991.3523	1320.4733
1	trip-153671042288605164	85.1110	84.1894
2	trip-153671043369099517	2372.0852	2545.2678
3	trip-153671046011330457	19.6800	19.8766
4	trip-153671052974046625	146.7918	146.7919
...	...	...	...
14812	trip-153861095625827784	73.4630	64.8551
14813	trip-153861104386292051	16.0882	16.0883
14814	trip-153861106442901555	63.2841	104.8866
14815	trip-153861115439069069	177.6635	223.5324
14816	trip-153861118270144424	80.5787	80.5787

14817 rows × 3 columns

In [66]:
```
mean_of_observation=np.mean(w['osrm_distance_agg'])-np.mean(w['segment_osrm_distance_agg'])
mean_of_observation
```

Out[66]:
```
-18.09980379968954
```

It implies that mean of 'osrm_distance_agg' not equal to 'segment_osrm_distance_agg'. So rejecting the null hypothesis

Do hypothesis testing/ visual analysis between osrm time aggregated value and segment osrm time aggregated value (aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)

In [67]:
```
w=df1.groupby(['trip_uuid','source_center']).agg({'osrm_time':'max','segment_osrm_time':'sum'})
w=w.groupby('trip_uuid').agg({'osrm_time':'sum','segment_osrm_time':'sum'}).reset_index()
w.rename(columns={'osrm_time':'osrm_time_agg','segment_osrm_time':'segment_osrm_time_agg'},inplace=True)
w
```

Out[67]:

	trip_uuid	osrm_time_agg	segment_osrm_time_agg
0	trip-153671041653548748	743.0	1008.0
1	trip-153671042288605164	68.0	65.0
2	trip-153671043369099517	1741.0	1941.0
3	trip-153671046011330457	15.0	16.0
4	trip-153671052974046625	117.0	115.0
...	...	...	...
14812	trip-153861095625827784	62.0	62.0
14813	trip-153861104386292051	12.0	11.0
14814	trip-153861106442901555	54.0	88.0
14815	trip-153861115439069069	184.0	221.0
14816	trip-153861118270144424	68.0	67.0

14817 rows × 3 columns

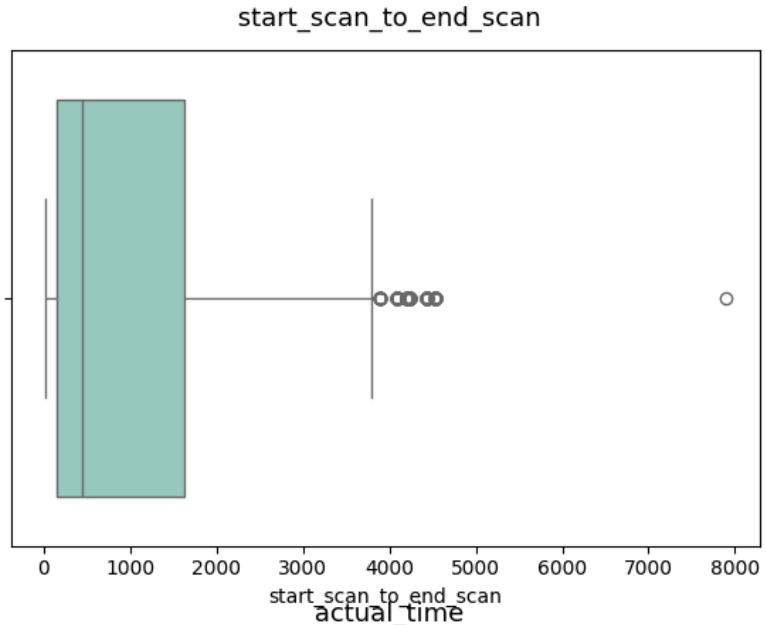
In [68]:
```
mean_observed_diff=np.mean(w['osrm_time_agg'])-np.mean(w['segment_osrm_time_agg'])
mean_observed_diff
```

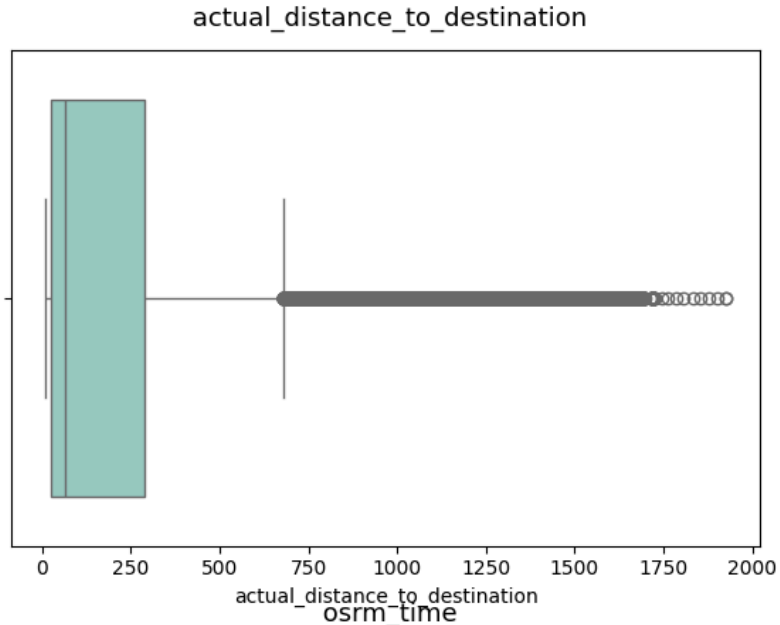
Out[68]:
```
-18.878585408652214
```

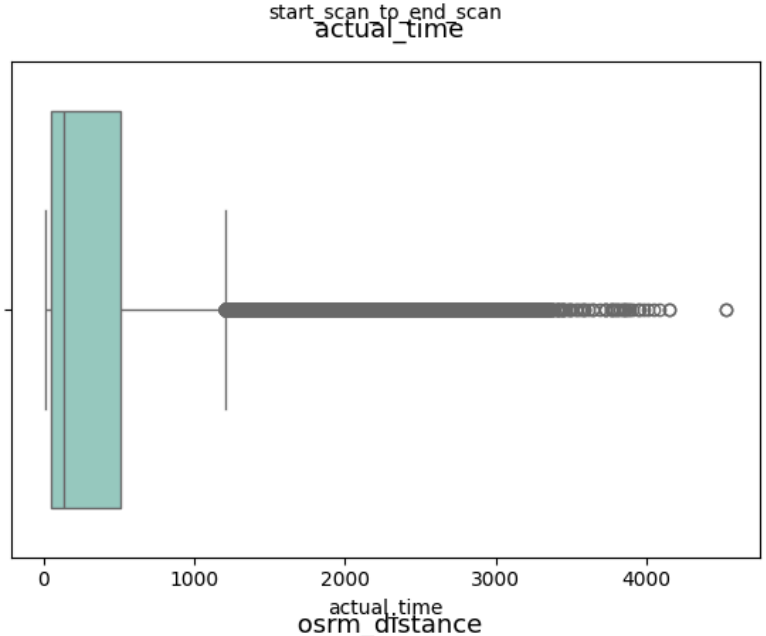
It implies that mean of 'osrm_time_agg' not equal to 'segment_osrm_time_agg'. So rejecting the null hypothesis

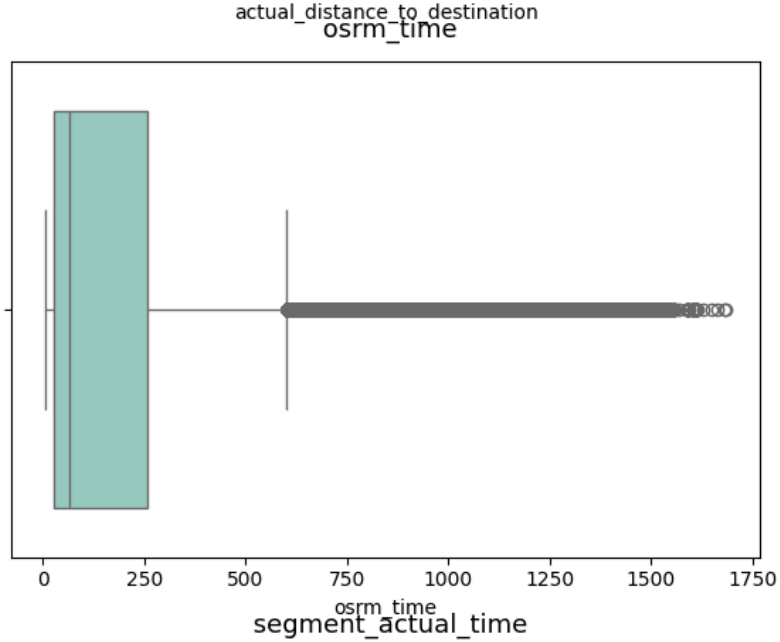
In [70]:
```
attrs=['start_scan_to_end_scan','actual_distance_to_destination','actual_time','osrm_time',
       'osrm_distance','segment_actual_time','segment_osrm_time','segment_osrm_distance']
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(15, 10))
```

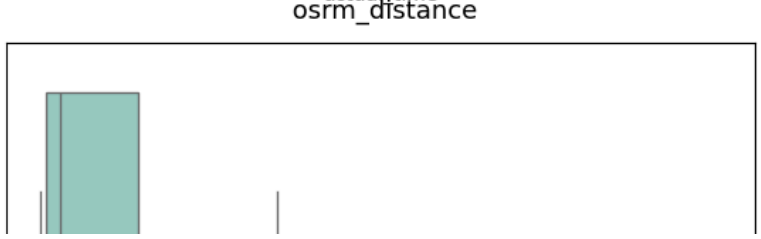
```
fig.subplots_adjust(top=1.3)
count = 0
for row in range(3):
    for col in range(2):
        sns.boxplot(data=df, x=attrs[count], ax=axs[row, col], palette='Set3')
        axs[row,col].set_title(f" {attrs[count]}", pad=12, fontsize=13)
        count += 1
plt.show()
```

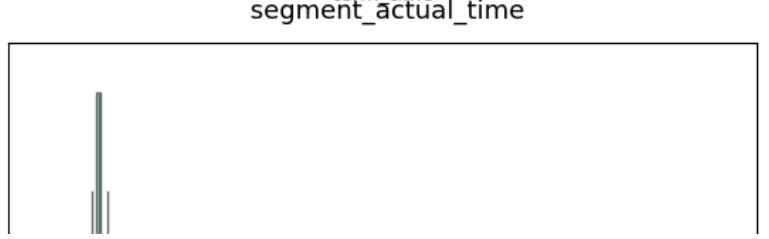

start_scan_to_end_scan
0
1000
2000
3000
4000
5000
6000
7000
8000
start_scan_to_end_scan

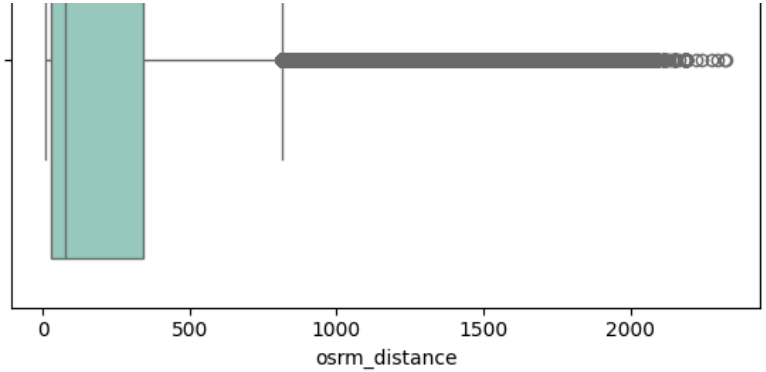

actual_distance_to_destination
0
250
500
750
1000
1250
1500
1750
2000
actual_distance_to_destination

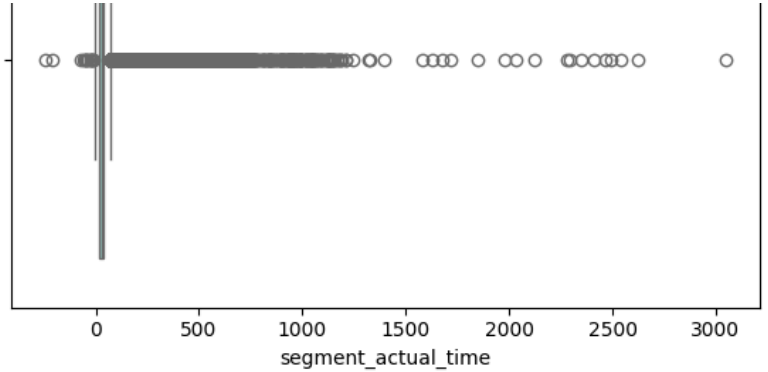

actual_time
0
1000
2000
3000
4000
actual_time


osrm_time
0
250
500
750
1000
1250
1500
1750
osrm_time


osrm_distance


segment_actual_time





In [73]:
```
#### Mean and median of attrs values shows large diffrance. So, we need to find outlayer values of purchace
ds=df.copy()
attrs=['start_scan_to_end_scan','actual_distance_to_destination','actual_time','osrm_time',
       'osrm_distance','segment_actual_time','segment_osrm_time','segment_osrm_distance']
for i in attrs:
    q1=df1[i].quantile(.25)
    q3=df1[i].quantile(.75)
    iqr=q3-q1
    lower=q1-(1.5*iqr)
    upper=q3+(1.5*iqr)
    print('lower limit of',i,'=',lower)
    print('upper limit of',i,'=',upper)
    print('------------------------------------------------------------------')
    ds=ds[~((ds[i]<lower)|(ds[i]>upper))]
ds
```

```
lower limit of start_scan_to_end_scan = -583.0
upper limit of start_scan_to_end_scan = 1369.0
------------------------------------------------------------
lower limit of actual_distance_to_destination = -190.12991073629408
upper limit of actual_distance_to_destination = 377.8432652176542
------------------------------------------------------------
lower limit of actual_time = -387.5
upper limit of actual_time = 824.5
------------------------------------------------------------
lower limit of osrm_time = -181.0
upper limit of osrm_time = 379.0
------------------------------------------------------------
lower limit of osrm_distance = -236.59625
upper limit of osrm_distance = 476.83855
------------------------------------------------------------
lower limit of segment_actual_time = -385.5
upper limit of segment_actual_time = 818.5
------------------------------------------------------------
lower limit of segment_osrm_time = -200.0
upper limit of segment_osrm_time = 416.0
------------------------------------------------------------
lower limit of segment_osrm_distance = -246.56735000000003
upper limit of segment_osrm_distance = 498.02425000000005
------------------------------------------------------------
```

Out[73]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_center	source_name
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)
...	...	...	...	...	...	...	...
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)

100641 rows × 24 columns

One-hot encoding of categorical variables (like route_type)

In [74]:
```
df_new = pd.get_dummies(ds, columns=["route_type"])
df_new
```

Out[74]:

	data	trip_creation_time	route_schedule_uuid	trip_uuid	source_center	source_name	destinatior
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388
...	...	...	...	...	...	...	
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000

100641 rows × 25 columns

Normalize/ Standardize the numerical features using MinMaxScaler or StandardScale

In [75]:
```
df_new=df_new.drop(['data','trip_creation_time','source_name','is_cutoff','destination_name','cutoff_factor','factor
            ,'segment_factor'],axis=1)
```

In [76]:
```
df_new
```

Out[76]:

	route_schedule_uuid	trip_uuid	source_center	destination_center	od_start_time	od_end_time	start_s
0	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	IND388620AAB	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	
1	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	IND388620AAB	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	
2	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	IND388620AAB	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	
3	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	IND388620AAB	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	
4	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	trip-153741093647649320	IND388121AAA	IND388620AAB	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	
...	...	...	...	...	...	...	
144862	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	IND000000ACB	2018-09-20 16:24:28.436231	2018-09-20 23:32:09.618069	
144863	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	IND000000ACB	2018-09-20 16:24:28.436231	2018-09-20 23:32:09.618069	
144864	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	IND000000ACB	2018-09-20 16:24:28.436231	2018-09-20 23:32:09.618069	
144865	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	IND000000ACB	2018-09-20 16:24:28.436231	2018-09-20 23:32:09.618069	
144866	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	trip-153746066843555182	IND131028AAB	IND000000ACB	2018-09-20 16:24:28.436231	2018-09-20 23:32:09.618069	

100641 rows × 16 columns

Recommendation and Insights

For deliver the products 60.12% prioritize carting shipments rout type because FTL shipments Full Truck Load takes more time to deliver product and time taken to actual and osrm higher than that of rout type Using carting shipments helps to reach destination sooner. Full truck load route type travels more distance to deliver products than that of the carting route type.so choosing cariting shipment is reduce time wasting Most of the products were delivered during wednesday followed by saturday and thursday. Least number of products were delivered on sunday.This indicate that it is preferable for customers to get productsdelivered during weekdays other than sundays or mondays The observed value for actual time is significantly higher than that of the estimated time for product delivery. Time taken for delivery and total trip time differ the most in Mizoram, Himachal Pradesh and Uttarakhand States like Dadra & Nagar Haveli and Tamil Nadu, the difference between actual and estimatedtime is low. Also product delivery in places like Mizoram and Himachal Pradesh, actual and osrm time difference are very high. Using routing machine give optimum results, then the difference can explained the traffic conditions or distance through major and minor roads Similarly the observed value for actual distance is always higher than that of the osrm distance for product delivery.Using An open-source routing engine which computes the shortest path between points,can explained the traffic conditions

Recommendations

Based on the insights for Delhivery's logistics operations, here are some recommendations to improve delivery efficiency:

Prioritize Carting Shipments Over FTL:

Since carting shipments have a shorter delivery time and cover less distance than FTL, Delhivery should prioritize carting for time-sensitive shipments. This will reduce time wastage and enhance delivery speed, especially for shorter routes or high-density areas. FTL could still be used for bulk deliveries or when shipping to distant, low-density locations where efficiency isn't solely dependent on speed. Optimizing Weekday Deliveries:

Focus delivery operations on peak delivery days (Wednesday, Saturday, and Thursday) and adjust logistics resources accordingly to handle higher demand on these days. Analyze why Sundays and Mondays have lower deliveries. Consider offering incentives for deliveries during these days to distribute the load more evenly across the week. Leverage Routing Engines:

Implement or enhance the use of open-source routing engines (like OSRM) to generate more accurate distance and time estimates. For areas with high discrepancies, integrate real-time traffic and weather data to improve routing accuracy. Consider dynamic routing that adjusts based on real-time conditions, especially in regions prone to traffic or adverse road conditions

Customer Communication:

Improve communication with customers in regions with longer delivery times, managing their expectations by providing more accurate delivery windows based on historical and real-time data.