cause-analysis-casestudy-kuntal-k

October 14, 2024

[7]:
```python
# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
import warnings
warnings.filterwarnings('ignore')
```

1 Data Observation

[9]:
```python
# Loading the dataset
dt = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/
 ↪000/001/551/original/delhivery_data.csv?1642751181")
dt.head(3)
```

[9]:
```
       data          trip_creation_time  \
0  training  2018-09-20 02:35:36.476840
1  training  2018-09-20 02:35:36.476840
2  training  2018-09-20 02:35:36.476840

                                 route_schedule_uuid route_type  \
0  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
1  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
2  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting

               trip_uuid source_center                 source_name  \
0  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
1  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
2  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)

  destination_center               destination_name  \
0       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
```

```
                 od_start_time  …            cutoff_timestamp  \
0  2018-09-20 03:21:32.418600  …         2018-09-20 04:27:55
1  2018-09-20 03:21:32.418600  …         2018-09-20 04:17:55
2  2018-09-20 03:21:32.418600  …  2018-09-20 04:01:19.505586

   actual_distance_to_destination  actual_time  osrm_time osrm_distance  \
0                       10.435660         14.0       11.0       11.9653
1                       18.936842         24.0       20.0       21.7243
2                       27.637279         40.0       28.0       32.5395

     factor  segment_actual_time  segment_osrm_time  segment_osrm_distance  \
0  1.272727                 14.0               11.0                11.9653
1  1.200000                 10.0                9.0                 9.7590
2  1.428571                 16.0                7.0                10.8152

   segment_factor
0        1.272727
1        1.111111
2        2.285714

[3 rows x 24 columns]
```

[10]:
```
# getting the counts of rows and columns in the dataset
dt.shape
```

[10]:
```
(144867, 24)
```

[11]:
```
# getting the information of the dataset
dt.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
```

2 Column Profiling:

data - tells whether the data is testing or training data
trip_creation_time – Timestamp of trip creation
route_schedule_uuid – Unique Id for a particular route schedule
route_type – Transportation type
FTL – Full Truck Load: FTL shipments get to the destination sooner, as the truck is making no other pickups or drop-offs along the way Carting: Handling system consisting of small vehicles (carts)
trip_uuid - Unique ID given to a particular trip (A trip may include different source and destination centers)
source_center - Source ID of trip origin
source_name - Source Name of trip origin
destination_cente – Destination ID
destination_name – Destination Name
od_start_time – Trip start time
od_end_time – Trip end time
start_scan_to_end_scan – Time taken to deliver from source to destination
is_cutoff – Unknown field
cutoff_factor – Unknown field
cutoff_timestamp – Unknown field
actual_distance_to_destination – Distance in Kms between source and destination warehouse
actual_time – Actual time taken to complete the delivery (Cumulative)
osrm_time – An open-source routing engine time calculator which computes the shortest path between points in a given map (Includes usual traffic, distance through major and minor roads) and gives the time (Cumulative)
osrm_distance – An open-source routing engine which computes the shortest path between points in a given map (Includes usual traffic, distance through major and minor roads) (Cumulative)
factor – Unknown field
segment_actual_time – This is a segment time. Time taken by the subset of the package delivery
segment_osrm_time – This is the OSRM segment time. Time taken by the subset of the pack-

age delivery
segment_osrm_distance – This is the OSRM distance. Distance covered by subset of the package delivery
segment_factor – Unknown field

```
[12]: # checking for null values
      dt.isnull().sum()
```

```
[12]: data                              0
      trip_creation_time                0
      route_schedule_uuid               0
      route_type                        0
      trip_uuid                         0
      source_center                     0
      source_name                     293
      destination_center                0
      destination_name                261
      od_start_time                     0
      od_end_time                       0
      start_scan_to_end_scan            0
      is_cutoff                         0
      cutoff_factor                     0
      cutoff_timestamp                  0
      actual_distance_to_destination    0
      actual_time                       0
      osrm_time                         0
      osrm_distance                     0
      factor                            0
      segment_actual_time               0
      segment_osrm_time                 0
      segment_osrm_distance             0
      segment_factor                    0
      dtype: int64
```

We can see that null values are present in source_name and destination_name column

```
[13]: # Removing rows if any null values are present
      data = dt.dropna(how = 'any')
```

```
[14]: # getting the rows and column no. of original table
      dt.shape
```

```
[14]: (144867, 24)
```

```
[15]: # getting the rows and column no. of new table after removing null values rows

      data.shape
```

```
[15]: (144316, 24)
```

```
[16]: # chcking the null values of nuw tables
      data.isnull().sum()
```

```
[16]: data                              0
      trip_creation_time                0
      route_schedule_uuid               0
      route_type                        0
      trip_uuid                         0
      source_center                     0
      source_name                       0
      destination_center                0
      destination_name                  0
      od_start_time                     0
      od_end_time                       0
      start_scan_to_end_scan            0
      is_cutoff                         0
      cutoff_factor                     0
      cutoff_timestamp                  0
      actual_distance_to_destination    0
      actual_time                       0
      osrm_time                         0
      osrm_distance                     0
      factor                            0
      segment_actual_time               0
      segment_osrm_time                 0
      segment_osrm_distance             0
      segment_factor                    0
      dtype: int64
```

```
[17]: # no of unique values in each column of new table
      for i in data.columns:
        print(i,data[i].nunique())
```

```
data 2
trip_creation_time 14787
route_schedule_uuid 1497
route_type 2
trip_uuid 14787
source_center 1496
source_name 1496
destination_center 1466
destination_name 1466
od_start_time 26223
od_end_time 26223
start_scan_to_end_scan 1914
```

```
is_cutoff 2
cutoff_factor 501
cutoff_timestamp 92894
actual_distance_to_destination 143965
actual_time 3182
osrm_time 1531
osrm_distance 137544
factor 45588
segment_actual_time 746
segment_osrm_time 214
segment_osrm_distance 113497
segment_factor 5663
```

[18]:
```
# coverting the object dtype time columns into pandas datetime dtype
data['od_start_time']  = pd.to_datetime(data['od_start_time'])
data['od_end_time'] = pd.to_datetime(data['od_end_time'])
data['trip_creation_time'] = pd.to_datetime(data['trip_creation_time'])
```

[19]:
```
# checking datatypes of new table
data.dtypes
```

[19]:
```
data                                      object
trip_creation_time                datetime64[ns]
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                     datetime64[ns]
od_end_time                       datetime64[ns]
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
```

[20]:
```python
# Get the unique values of the 'data' column
unique_data = data['data'].unique()
unique_data
```

[20]:
```
array(['training', 'test'], dtype=object)
```

[21]:
```python
# Get the unique values of the 'route_type' column
unique_route_type = data['route_type'].unique()
unique_route_type
```

[21]:
```
array(['Carting', 'FTL'], dtype=object)
```

[22]:
```python
# creating a new new col by merging 3 columns
data['segment_key'] =data['trip_uuid'] + data['source_center'] +␣
 ↪data['destination_center']
data['segment_key'].head(3)
```

[22]:
```
0    trip-153741093647649320IND388121AAAIND388620AAB
1    trip-153741093647649320IND388121AAAIND388620AAB
2    trip-153741093647649320IND388121AAAIND388620AAB
Name: segment_key, dtype: object
```

[23]:
```python
# creating a df segment_col of 3 columns
segment_col =␣
 ↪['segment_actual_time','segment_osrm_time','segment_osrm_distance']
data[segment_col].head()
```

[23]:
```
   segment_actual_time  segment_osrm_time  segment_osrm_distance
0                 14.0               11.0                11.9653
1                 10.0                9.0                 9.7590
2                 16.0                7.0                10.8152
3                 21.0               12.0                13.0224
4                  6.0                5.0                 3.9153
```

[24]:
```python
# Grouping by the segment_key of sub-journey in the trip
for i in segment_col:
  data[i+'_sum'] = data.groupby('segment_key')[i].cumsum()

segment_col_sum = [i+'_sum' for i in segment_col]
data[segment_col_sum].head()
```

[24]:
```
   segment_actual_time_sum  segment_osrm_time_sum  segment_osrm_distance_sum
0                     14.0                   11.0                    11.9653
1                     24.0                   20.0                    21.7243
2                     40.0                   27.0                    32.5395
3                     61.0                   39.0                    45.5619
4                     67.0                   44.0                    49.4772
```

[25]:
```
# creating a dictionary of 19 columns for aggreating at sub sourney level
agg_dict = {'data': 'first', 'trip_creation_time': 'first',␣
↪'route_schedule_uuid': 'first', 'route_type': 'first',
            'trip_uuid' : 'first', 'source_center' : 'first', 'source_name' :␣
↪'first', 'destination_center' : 'last',
            'destination_name' : 'last', 'od_start_time' : 'first',␣
↪'od_end_time' : 'first', 'start_scan_to_end_scan' : 'first',
            'actual_distance_to_destination' : 'last', 'actual_time' : 'last',␣
↪'osrm_time' : 'last', 'osrm_distance' : 'last',
            'segment_actual_time_sum' : 'last', 'segment_osrm_distance_sum' :␣
↪'last', 'segment_osrm_time_sum' : 'last',}
agg_dict
```

[25]:
```
{'data': 'first',
 'trip_creation_time': 'first',
 'route_schedule_uuid': 'first',
 'route_type': 'first',
 'trip_uuid': 'first',
 'source_center': 'first',
 'source_name': 'first',
 'destination_center': 'last',
 'destination_name': 'last',
 'od_start_time': 'first',
 'od_end_time': 'first',
 'start_scan_to_end_scan': 'first',
 'actual_distance_to_destination': 'last',
 'actual_time': 'last',
 'osrm_time': 'last',
 'osrm_distance': 'last',
 'segment_actual_time_sum': 'last',
 'segment_osrm_distance_sum': 'last',
 'segment_osrm_time_sum': 'last'}
```

[26]:
```
segment = data.groupby('segment_key').agg(agg_dict).reset_index()
segment.head()
```

[26]:
```
                                       segment_key      data  \
0  trip-153671041653548748IND209304AAAIND000000ACB  training
1  trip-153671041653548748IND462022AAAIND209304AAA  training
2  trip-153671042288605164IND561203AABIND562101AAA  training
3  trip-153671042288605164IND572101AAAIND561203AAB  training
4  trip-153671043369099517IND000000ACBIND160002AAC  training

          trip_creation_time  \
0 2018-09-12 00:00:16.535741
1 2018-09-12 00:00:16.535741
2 2018-09-12 00:00:22.886430
```

```
3 2018-09-12 00:00:22.886430
4 2018-09-12 00:00:33.691250

                               route_schedule_uuid route_type  \
0  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
1  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
2  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
3  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
4  thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e…        FTL

               trip_uuid source_center                          source_name  \
0  trip-153671041653548748  IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1  trip-153671041653548748  IND462022AAA  Bhopal_Trnsport_H (Madhya Pradesh)
2  trip-153671042288605164  IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
3  trip-153671042288605164  IND572101AAA       Tumkur_Veersagr_I (Karnataka)
4  trip-153671043369099517  IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)

  destination_center                    destination_name  \
0       IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)
1       IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
2       IND562101AAA   Chikblapur_ShntiSgr_D (Karnataka)
3       IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
4       IND160002AAC      Chandigarh_Mehmdpur_H (Punjab)

               od_start_time                od_end_time  \
0 2018-09-12 16:39:46.858469 2018-09-13 13:40:23.123744
1 2018-09-12 00:00:16.535741 2018-09-12 16:39:46.858469
2 2018-09-12 02:03:09.655591 2018-09-12 03:01:59.598855
3 2018-09-12 00:00:22.886430 2018-09-12 02:03:09.655591
4 2018-09-14 03:40:17.106733 2018-09-14 17:34:55.442454

   start_scan_to_end_scan  actual_distance_to_destination  actual_time  \
0                  1260.0                      383.759164        732.0
1                   999.0                      440.973689        830.0
2                    58.0                       24.644021         47.0
3                   122.0                       48.542890         96.0
4                   834.0                      237.439610        611.0

   osrm_time  osrm_distance  segment_actual_time_sum  \
0      329.0       446.5496                    728.0
1      388.0       544.8027                    820.0
2       26.0        28.1994                     46.0
3       42.0        56.9116                     95.0
4      212.0       281.2109                    608.0

   segment_osrm_distance_sum  segment_osrm_time_sum
0                   670.6205                  534.0
```

```
1                  649.8528                 474.0
2                   28.1995                  26.0
3                   55.9899                  39.0
4                  317.7408                 231.0
```

[27]:
```
#Groupby mini-trips, sorting by time
segment  = segment.sort_values(by=['segment_key','od_end_time'],␣
 ↪ascending=True).reset_index()
segment.head(3)
```

[27]:
```
   index                                     segment_key      data  \
0      0  trip-153671041653548748IND209304AAAIND000000ACB  training
1      1  trip-153671041653548748IND462022AAAIND209304AAA  training
2      2  trip-153671042288605164IND561203AABIND562101AAA  training

         trip_creation_time  \
0 2018-09-12 00:00:16.535741
1 2018-09-12 00:00:16.535741
2 2018-09-12 00:00:22.886430


                               route_schedule_uuid route_type  \
0  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
1  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
2  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting

                 trip_uuid source_center                          source_name  \
0  trip-153671041653548748  IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1  trip-153671041653548748  IND462022AAA  Bhopal_Trnsport_H (Madhya Pradesh)
2  trip-153671042288605164  IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)

  destination_center  …               od_start_time  \
0       IND000000ACB  … 2018-09-12 16:39:46.858469
1       IND209304AAA  … 2018-09-12 00:00:16.535741
2       IND562101AAA  … 2018-09-12 02:03:09.655591

                od_end_time start_scan_to_end_scan  \
0 2018-09-13 13:40:23.123744                 1260.0
1 2018-09-12 16:39:46.858469                  999.0
2 2018-09-12 03:01:59.598855                   58.0

   actual_distance_to_destination  actual_time  osrm_time  osrm_distance  \
0                      383.759164        732.0      329.0       446.5496
1                      440.973689        830.0      388.0       544.8027
2                       24.644021         47.0       26.0        28.1994

   segment_actual_time_sum  segment_osrm_distance_sum  segment_osrm_time_sum
0                    728.0                   670.6205                  534.0
```

```
1                    820.0                     649.8528                    474.0
2                     46.0                      28.1995                     26.0

[3 rows x 21 columns]
```

[28]:
```
# getting info of one trip id from segment table
segment[segment['trip_uuid'] == 'trip-153671041653548748']
```

[28]:
```
   index                                    segment_key      data  \
0      0  trip-153671041653548748IND209304AAAIND000000ACB  training
1      1  trip-153671041653548748IND462022AAAIND209304AAA  training

          trip_creation_time  \
0 2018-09-12 00:00:16.535741
1 2018-09-12 00:00:16.535741

                               route_schedule_uuid route_type  \
0  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
1  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL

                trip_uuid source_center                         source_name  \
0  trip-153671041653548748  IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1  trip-153671041653548748  IND462022AAA  Bhopal_Trnsport_H (Madhya Pradesh)

  destination_center  …              od_start_time  \
0       IND000000ACB  … 2018-09-12 16:39:46.858469
1       IND209304AAA  … 2018-09-12 00:00:16.535741

                od_end_time start_scan_to_end_scan  \
0 2018-09-13 13:40:23.123744                 1260.0
1 2018-09-12 16:39:46.858469                  999.0

   actual_distance_to_destination  actual_time  osrm_time  osrm_distance  \
0                      383.759164        732.0      329.0       446.5496
1                      440.973689        830.0      388.0       544.8027

   segment_actual_time_sum  segment_osrm_distance_sum  segment_osrm_time_sum
0                    728.0                   670.6205                  534.0
1                    820.0                   649.8528                  474.0

[2 rows x 21 columns]
```

[29]:
```
# getting the info of segment table
segment.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26222 entries, 0 to 26221
```

```
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   index                           26222 non-null  int64
 1   segment_key                     26222 non-null  object
 2   data                            26222 non-null  object
 3   trip_creation_time              26222 non-null  datetime64[ns]
 4   route_schedule_uuid             26222 non-null  object
 5   route_type                      26222 non-null  object
 6   trip_uuid                       26222 non-null  object
 7   source_center                   26222 non-null  object
 8   source_name                     26222 non-null  object
 9   destination_center              26222 non-null  object
 10  destination_name                26222 non-null  object
 11  od_start_time                   26222 non-null  datetime64[ns]
 12  od_end_time                     26222 non-null  datetime64[ns]
 13  start_scan_to_end_scan          26222 non-null  float64
 14  actual_distance_to_destination  26222 non-null  float64
 15  actual_time                     26222 non-null  float64
 16  osrm_time                       26222 non-null  float64
 17  osrm_distance                   26222 non-null  float64
 18  segment_actual_time_sum         26222 non-null  float64
 19  segment_osrm_distance_sum       26222 non-null  float64
 20  segment_osrm_time_sum           26222 non-null  float64
dtypes: datetime64[ns](3), float64(8), int64(1), object(9)
memory usage: 4.2+ MB
```

2.1 Feature Creation

2.1.1 Calculate time taken between od_start_time and od_end_time

[30]:
```
#getting the time difference between od_start_time and od_end_time in hour in␣
↪an separate column in segment table
segment['hour_taken'] = (segment['od_end_time'] - segment['od_start_time']).dt.
↪total_seconds()/60
segment['hour_taken'].head(3)
```

[30]:
```
0    1260.604421
1     999.505379
2      58.832388
Name: hour_taken, dtype: float64
```

[31]:
```
segment.head()
```

[31]:
```
   index                                   segment_key      data  \
0      0  trip-153671041653548748IND209304AAAIND000000ACB  training
1      1  trip-153671041653548748IND462022AAAIND209304AAA  training
2      2  trip-153671042288605164IND561203AABIND562101AAA  training
```

```
3      3  trip-153671042288605164IND572101AAAIND561203AAB  training
4      4  trip-153671043369099517IND000000ACBIND160002AAC  training

          trip_creation_time  \
0 2018-09-12 00:00:16.535741
1 2018-09-12 00:00:16.535741
2 2018-09-12 00:00:22.886430
3 2018-09-12 00:00:22.886430
4 2018-09-12 00:00:33.691250

                              route_schedule_uuid route_type  \
0  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
1  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
2  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
3  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
4  thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e…        FTL

                trip_uuid source_center                          source_name  \
0  trip-153671041653548748  IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1  trip-153671041653548748  IND462022AAA  Bhopal_Trnsport_H (Madhya Pradesh)
2  trip-153671042288605164  IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
3  trip-153671042288605164  IND572101AAA       Tumkur_Veersagr_I (Karnataka)
4  trip-153671043369099517  IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)

  destination_center  …                od_end_time start_scan_to_end_scan  \
0       IND000000ACB  … 2018-09-13 13:40:23.123744                 1260.0
1       IND209304AAA  … 2018-09-12 16:39:46.858469                  999.0
2       IND562101AAA  … 2018-09-12 03:01:59.598855                   58.0
3       IND561203AAB  … 2018-09-12 02:03:09.655591                  122.0
4       IND160002AAC  … 2018-09-14 17:34:55.442454                  834.0

  actual_distance_to_destination  actual_time  osrm_time  osrm_distance  \
0                     383.759164        732.0      329.0       446.5496
1                     440.973689        830.0      388.0       544.8027
2                      24.644021         47.0       26.0        28.1994
3                      48.542890         96.0       42.0        56.9116
4                     237.439610        611.0      212.0       281.2109

   segment_actual_time_sum  segment_osrm_distance_sum  segment_osrm_time_sum  \
0                    728.0                   670.6205                  534.0
1                    820.0                   649.8528                  474.0
2                     46.0                    28.1995                   26.0
3                     95.0                    55.9899                   39.0
4                    608.0                   317.7408                  231.0

    hour_taken
0  1260.604421
```

```
1    999.505379
2     58.832388
3    122.779486
4    834.638929

[5 rows x 22 columns]
```

[32]:
```
# creating another dictionary for aggregation in gesment table
trip_dict = {'data' : 'first', 'trip_creation_time': 'first',␣
 ↪'route_schedule_uuid' : 'first', 'route_type' : 'first','trip_uuid' :␣
 ↪'first', 'source_center' : 'first', 'source_name' : 'first',
             'destination_center' : 'last', 'destination_name' : 'last',␣
 ↪'start_scan_to_end_scan' : 'sum', 'hour_taken' : 'sum',␣
 ↪'actual_distance_to_destination' : 'sum',
             'actual_time' : 'sum', 'osrm_time' : 'sum', 'osrm_distance' :␣
 ↪'sum', 'segment_actual_time_sum' : 'sum', 'segment_osrm_distance_sum' :␣
 ↪'sum', 'segment_osrm_time_sum' : 'sum',}

trip_dict
```

[32]:
```
{'data': 'first',
 'trip_creation_time': 'first',
 'route_schedule_uuid': 'first',
 'route_type': 'first',
 'trip_uuid': 'first',
 'source_center': 'first',
 'source_name': 'first',
 'destination_center': 'last',
 'destination_name': 'last',
 'start_scan_to_end_scan': 'sum',
 'hour_taken': 'sum',
 'actual_distance_to_destination': 'sum',
 'actual_time': 'sum',
 'osrm_time': 'sum',
 'osrm_distance': 'sum',
 'segment_actual_time_sum': 'sum',
 'segment_osrm_distance_sum': 'sum',
 'segment_osrm_time_sum': 'sum'}
```

[88]:
```
# creating another table, groupping by trip id in segment table
trip = segment.groupby('trip_uuid').agg(trip_dict)
trip = trip.reset_index(drop = True)
trip.head()
```

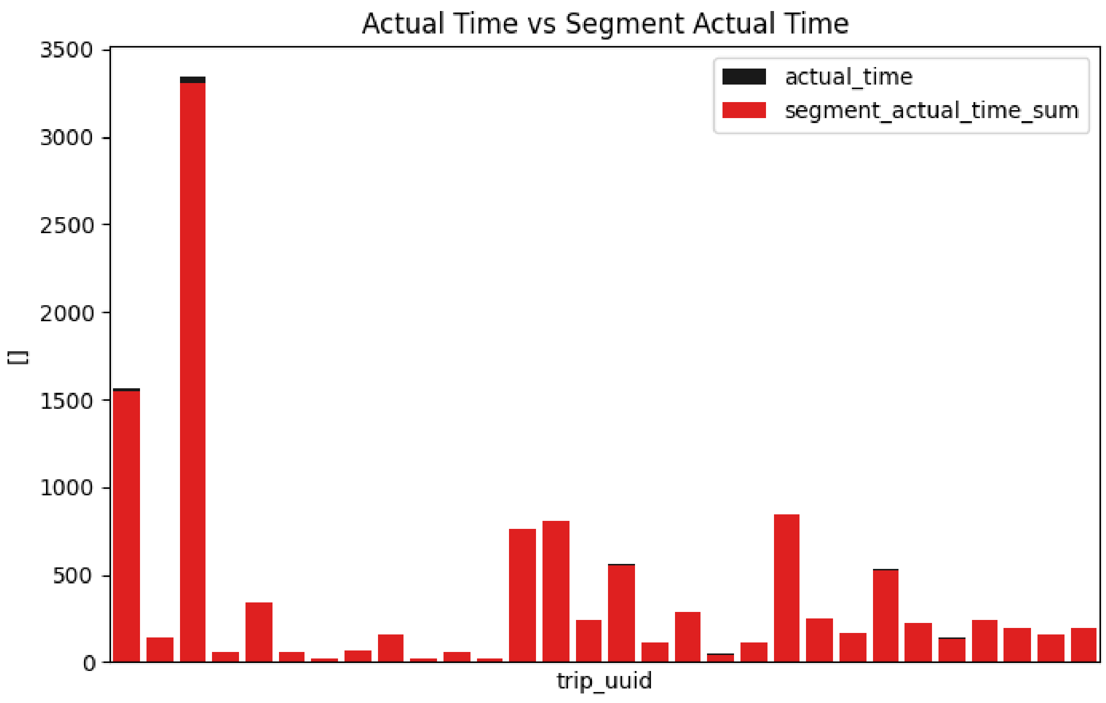
[88]:
```
       data          trip_creation_time  \
0  training 2018-09-12 00:00:16.535741
1  training 2018-09-12 00:00:22.886430
```

```
2  training 2018-09-12 00:00:33.691250
3  training 2018-09-12 00:01:00.113710
4  training 2018-09-12 00:02:09.740725

                              route_schedule_uuid route_type  \
0  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
1  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
2  thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e…        FTL
3  thanos::sroute:f0176492-a679-4597-8332-bbd1c7f…    Carting
4  thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134…        FTL

               trip_uuid source_center                          source_name  \
0  trip-153671041653548748  IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1  trip-153671042288605164  IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
2  trip-153671043369099517  IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)
3  trip-153671046011330457  IND400072AAB            Mumbai Hub (Maharashtra)
4  trip-153671052974046625  IND583101AAA              Bellary_Dc (Karnataka)

  destination_center                    destination_name  \
0       IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1       IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
2       IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)
3       IND401104AAA      Mumbai_MiraRd_IP (Maharashtra)
4       IND583119AAA       Sandur_WrdN1DPP_D (Karnataka)

   start_scan_to_end_scan   hour_taken  actual_distance_to_destination  \
0                  2259.0  2260.109800                      824.732854
1                   180.0   181.611874                       73.186911
2                  3933.0  3934.362520                     1927.404273
3                   100.0   100.494935                       17.175274
4                   717.0   718.349042                      127.448500

   actual_time  osrm_time  osrm_distance  segment_actual_time_sum  \
0       1562.0      717.0       991.3523                   1548.0
1        143.0       68.0        85.1110                    141.0
2       3347.0     1740.0      2354.0665                   3308.0
3         59.0       15.0        19.6800                     59.0
4        341.0      117.0       146.7918                    340.0

   segment_osrm_distance_sum  segment_osrm_time_sum
0                  1320.4733                 1008.0
1                    84.1894                   65.0
2                  2545.2678                 1941.0
3                    19.8766                   16.0
4                   146.7919                  115.0
```

```
[34]: trip1 = trip[['actual_time', 'segment_actual_time_sum','trip_uuid']].head(30)
      trip1.head(3)
```

```
[34]:    actual_time  segment_actual_time_sum                trip_uuid
      0       1562.0                   1548.0  trip-153671041653548748
      1        143.0                    141.0  trip-153671042288605164
      2       3347.0                   3308.0  trip-153671043369099517
```

```
[35]: # Create the figure with a transparent background
      plt.figure(figsize=(8, 5), facecolor='none')

      sns.barplot(data=trip1, y = 'actual_time', x = 'trip_uuid', label =␣
       ↪'actual_time', color = 'black', alpha = 0.9)
      sns.barplot(data=trip1, y = 'segment_actual_time_sum', x = 'trip_uuid', color =␣
       ↪'red', label = 'segment_actual_time_sum')
      plt.title('Actual Time vs Segment Actual Time')
      # Remove x-axis labels
      plt.xticks([])
      plt.ylabel([])

      # Show the plot
      plt.show()
```



As we can see from the trend, ‘actual_time’and ’segment_actual_time_sum’of every ’trip_uuid’ are not that different

[36]:
```
trip.head(3)
```

[36]:
```
       data          trip_creation_time  \
0  training 2018-09-12 00:00:16.535741
1  training 2018-09-12 00:00:22.886430
2  training 2018-09-12 00:00:33.691250

                                 route_schedule_uuid route_type  \
0  thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6…        FTL
1  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
2  thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e…        FTL

                 trip_uuid source_center                         source_name  \
0  trip-153671041653548748  IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1  trip-153671042288605164  IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
2  trip-153671043369099517  IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)

  destination_center                    destination_name  \
0       IND209304AAA  Kanpur_Central_H_6 (Uttar Pradesh)
1       IND561203AAB   Doddablpur_ChikaDPP_D (Karnataka)
2       IND000000ACB       Gurgaon_Bilaspur_HB (Haryana)

   start_scan_to_end_scan   hour_taken  actual_distance_to_destination  \
0                  2259.0  2260.109800                      824.732854
1                   180.0   181.611874                       73.186911
2                  3933.0  3934.362520                     1927.404273

   actual_time  osrm_time  osrm_distance  segment_actual_time_sum  \
0       1562.0      717.0       991.3523                   1548.0
1        143.0       68.0        85.1110                    141.0
2       3347.0     1740.0      2354.0665                   3308.0

   segment_osrm_distance_sum  segment_osrm_time_sum
0                  1320.4733                 1008.0
1                    84.1894                   65.0
2                  2545.2678                 1941.0
```

[37]:
```
# checking the one trip id if from trip table
trip[trip['trip_uuid']=='trip-153671042288605164']
```

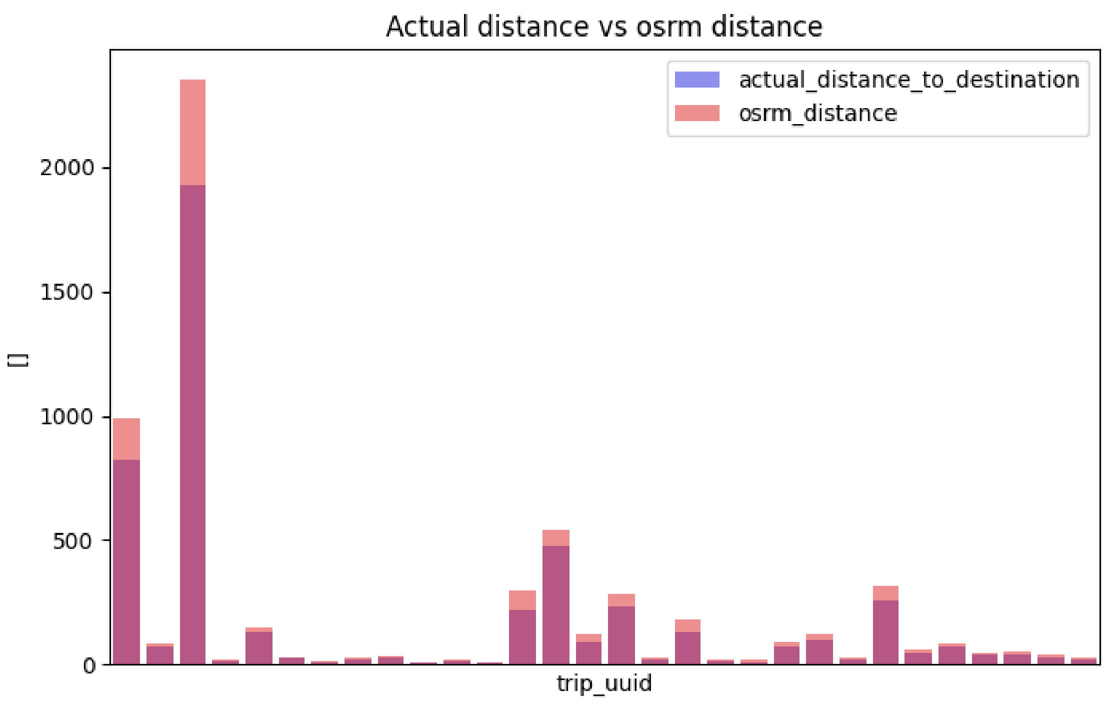
[37]:
```
       data          trip_creation_time  \
1  training 2018-09-12 00:00:22.886430

                                 route_schedule_uuid route_type  \
1  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
```

```
                 trip_uuid source_center                          source_name  \
1  trip-153671042288605164  IND561203AAB  Doddablpur_ChikaDPP_D (Karnataka)

  destination_center                   destination_name  \
1       IND561203AAB  Doddablpur_ChikaDPP_D (Karnataka)

   start_scan_to_end_scan  hour_taken  actual_distance_to_destination  \
1                   180.0  181.611874                       73.186911

   actual_time  osrm_time  osrm_distance  segment_actual_time_sum  \
1        143.0       68.0         85.111                    141.0

   segment_osrm_distance_sum  segment_osrm_time_sum
1                    84.1894                   65.0
```

[38]:
```
trip2 = trip[['actual_distance_to_destination','osrm_distance', 'trip_uuid']].
 ↪head(30)
trip2.head(3)
```

[38]:
```
   actual_distance_to_destination  osrm_distance                trip_uuid
0                      824.732854       991.3523  trip-153671041653548748
1                       73.186911        85.1110  trip-153671042288605164
2                     1927.404273      2354.0665  trip-153671043369099517
```

[39]:
```
# Create the figure with a transparent background
plt.figure(figsize=(8, 5), facecolor='none')

sns.barplot(data=trip2, y = 'actual_distance_to_destination', x = 'trip_uuid',␣
 ↪label = 'actual_distance_to_destination', color = 'blue', alpha = 0.5)
sns.barplot(data=trip2, y = 'osrm_distance', x = 'trip_uuid', color = 'red',␣
 ↪label = 'osrm_distance', alpha = 0.5)
plt.title('Actual distance vs osrm distance')
# Remove x-axis labels
plt.xticks([])
plt.ylabel([])

# Show the plot
plt.show()
```



2.2 ### as we can observe that there are slight differences between ‘actual_distance_to_destination’ and ‘osrm_distance’

[40]:
```
# converting all the values of destination_name and source_name column in lower␣
↪case

trip['destination_name'] = trip['destination_name'].str.lower()
trip['source_name'] = trip['source_name'].str.lower()
trip['source_name'].head(3)
```

[40]:
```
0    kanpur_central_h_6 (uttar pradesh)
1     doddablpur_chikadpp_d (karnataka)
2         gurgaon_bilaspur_hb (haryana)
Name: source_name, dtype: object
```

2.2.1 creating function to get the proper names from the columns

[41]:
```
def get_state(x):
  state = x.split('(')[1] # getting 'uttar pradesh)' from 'kanpur_central_h_6␣
↪(uttar pradesh)'

  return state[:-1] # returning 'uttar pradesh' from 'uttar pradesh)'
```

[42]:
```
def get_city(x):
  city = x.split('(')[0] # getting 'kanpur_central_h_6' from␣
  ↪'kanpur_central_h_6 (uttar pradesh)'
  city = city.split('_')[0] # getting 'kanpur'
      #Now dealing with edge cases
  if city == 'pnq vadgaon sheri dpc':
    return 'vadgaonsheri'

    # ['PNQ Pashan DPC', 'Bhopal MP Nagar', 'HBR Layout PC',
    #  'PNQ Rahatani DPC', 'Pune Balaji Nagar', 'Mumbai Antop Hill']

    if city in ['pnq pashan dpc','pnq rahatani dpc', 'pune balaji nagar']:
        return 'pune'

    if city == 'hbr layout pc' : return 'bengaluru'
    if city == 'bhopal mp nagar' : return 'bhopal'
    if city == 'mumbai antop hill' : return 'mumbai'

  return city
```

[43]:
```
def place2city_place(x):
    # We will remove state
    x = x.split(' (')[0]
    len_ = len(x.split('_'))
    if len_ >= 3:
        return x.split('_')[1]

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

[44]:
```
def get_code(x):
    # We will remove state
    x = x.split(' (')[0]

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

    return 'none'
```

[45]:
```
# getting the separate column of place , city, state , code from the␣
 ↪destination name
trip['destination_state'] = trip['destination_name'].apply(lambda x:␣
 ↪get_state(x))
trip['destination_city']  = trip['destination_name'].apply(lambda x:␣
 ↪get_city(x))
```

```
trip['destination_place'] = trip['destination_name'].apply(lambda x:␣
↪place2city_place(x))
trip['destination_code']  = trip['destination_name'].apply(lambda x:␣
↪get_code(x))

trip[['destination_state', 'destination_city', 'destination_place',␣
↪'destination_code']]
```

[45]:
```
      destination_state destination_city destination_place destination_code
0         uttar pradesh           kanpur           central                6
1             karnataka       doddablpur          chikadpp                d
2               haryana          gurgaon          bilaspur               hb
3           maharashtra           mumbai            mirard               ip
4             karnataka           sandur           wrdn1dpp               d
…                   …              …                 …              …
14782            punjab       chandigarh          mehmdpur                h
14783           haryana        faridabad           blbgarh               dc
14784     uttar pradesh           kanpur          govndngr               dc
14785        tamil nadu       tirchchndr          shnmgprm                d
14786         karnataka           sandur           wrdn1dpp               d

[14787 rows x 4 columns]
```

[46]:
```
# getting the separate column of place , city, state , code from the source name
trip['source_state'] = trip['source_name'].apply(lambda x: get_state(x))
trip['source_city']  = trip['source_name'].apply(lambda x: get_city(x))
trip['source_place'] = trip['source_name'].apply(lambda x: place2city_place(x))
trip['source_code']  = trip['source_name'].apply(lambda x: get_code(x))

trip[['source_state', 'source_city', 'source_place', 'source_code']]
```

[46]:
```
         source_state  source_city source_place source_code
0       uttar pradesh       kanpur      central           6
1           karnataka   doddablpur     chikadpp           d
2             haryana      gurgaon     bilaspur          hb
3         maharashtra  mumbai hub        mumbai        none
4           karnataka      bellary      bellary        none
…                 …            …            …          …
14782          punjab   chandigarh     mehmdpur           h
14783         haryana          fbd   balabhgarh         dpc
14784   uttar pradesh       kanpur     govndngr          dc
14785      tamil nadu  tirunelveli     vdkkusrt           i
14786       karnataka       sandur     wrdn1dpp           d

[14787 rows x 4 columns]
```

Univariate - Categorical Data

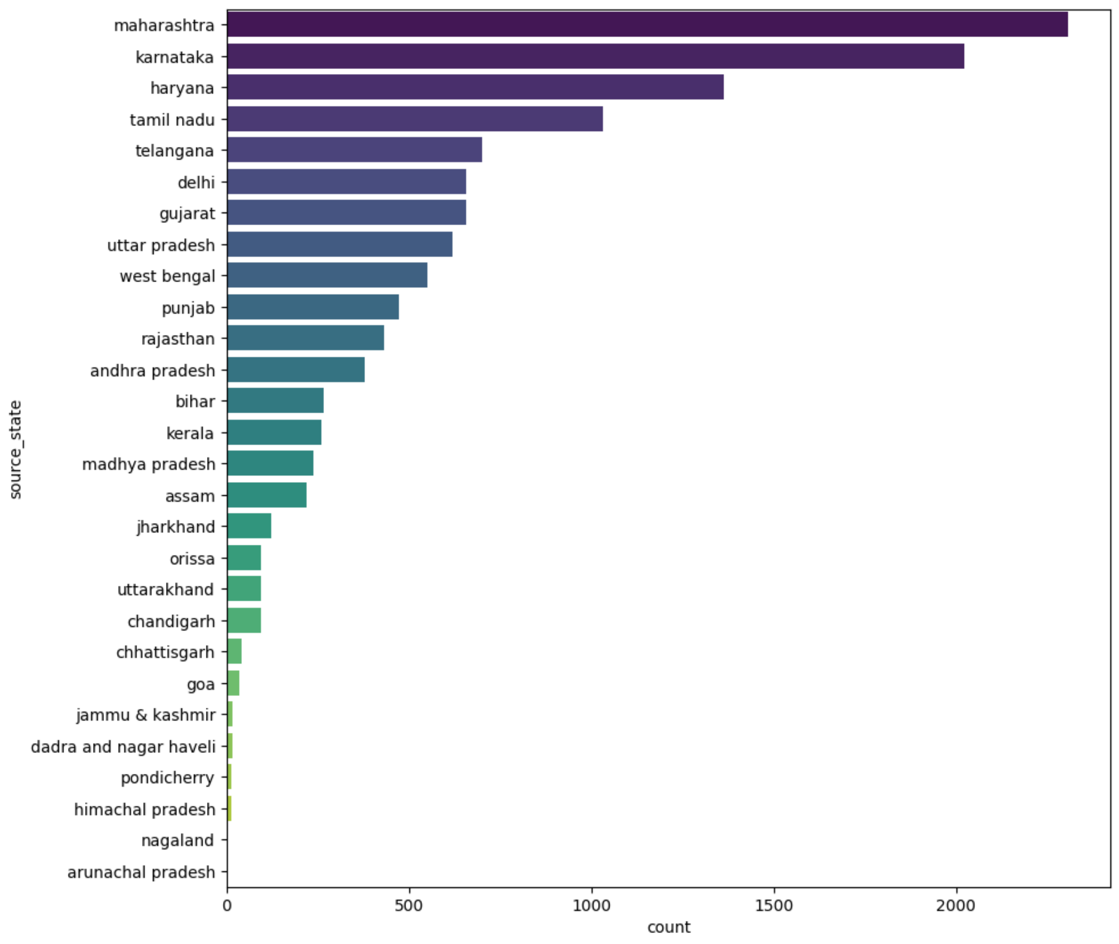
[77]:
```
trip_states = trip[['source_state']].value_counts().reset_index()
trip_states
```

[77]:
```
                source_state  count
0                maharashtra   2308
1                  karnataka   2025
2                    haryana   1365
3                 tamil nadu   1032
4                  telangana    701
5                      delhi    658
6                    gujarat    656
7              uttar pradesh    619
8                west bengal    551
9                     punjab    472
10                 rajasthan    431
11            andhra pradesh    378
12                     bihar    267
13                    kerala    261
14            madhya pradesh    238
15                     assam    220
16                 jharkhand    123
17                    orissa     94
18               uttarakhand     93
19                chandigarh     93
20              chhattisgarh     42
21                       goa     34
22           jammu & kashmir     16
23    dadra and nagar haveli     15
24               pondicherry     12
25          himachal pradesh     12
26                  nagaland      4
27         arunachal pradesh      3
```

Visualization of above chart

[81]:
```
plt.figure(figsize=(10,10))
sns.barplot(data = trip_states, y = 'source_state', x = 'count',␣
 ↪palette='viridis')
```

[81]:
```
<Axes: xlabel='count', ylabel='source_state'>
```



From the above chart, we get to know that from Maharashtra, maximum trip starts. North, South and West Zones corridors have significant traffic of orders. But, have a smaller presence in Central, Eastern and North-Eastern zone

[47]:
```
# getting the separate columns of year, month, hour, day , week, day of week,␣
↪from trip_creation_time
trip['trip_year'] = trip['trip_creation_time'].dt.year
trip['trip_month'] = trip['trip_creation_time'].dt.month
trip['trip_hour'] = trip['trip_creation_time'].dt.hour
trip['trip_day'] = trip['trip_creation_time'].dt.day
trip['trip_week'] = trip['trip_creation_time'].dt.isocalendar().week
trip['trip_dayofweek'] = trip['trip_creation_time'].dt.dayofweek

trip[['trip_year', 'trip_month', 'trip_hour', 'trip_day', 'trip_week',␣
↪'trip_dayofweek']]
```

[47]:
```
      trip_year  trip_month  trip_hour  trip_day  trip_week  trip_dayofweek
0          2018           9          0        12         37               2
```

```
1           2018        9       0       12      37          2
2           2018        9       0       12      37          2
3           2018        9       0       12      37          2
4           2018        9       0       12      37          2
…           …       …       …       …       …          …
14782       2018       10      23        3      40          2
14783       2018       10      23        3      40          2
14784       2018       10      23        3      40          2
14785       2018       10      23        3      40          2
14786       2018       10      23        3      40          2

[14787 rows x 6 columns]
```

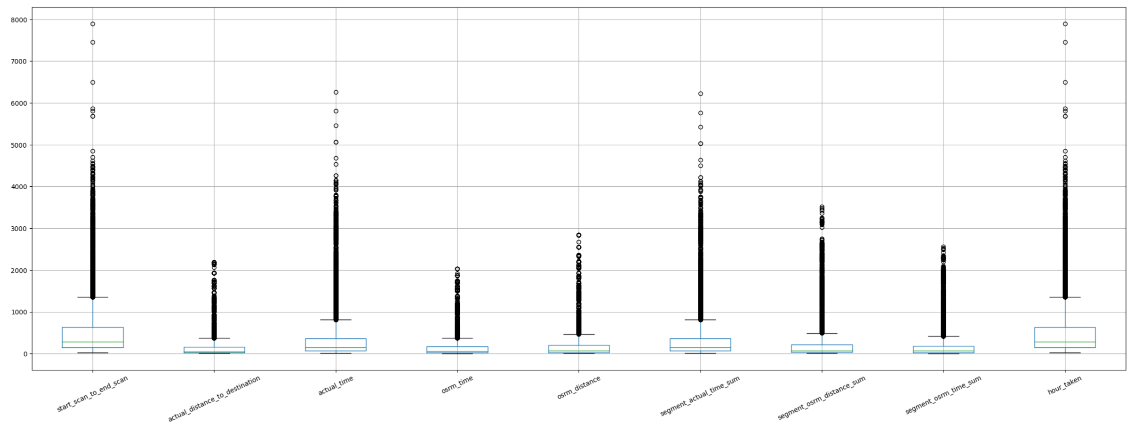
[48]:
```
# numerical columns
num_col = ['start_scan_to_end_scan', 'actual_distance_to_destination',␣
 ↪'actual_time', 'osrm_time',
            'osrm_distance', 'segment_actual_time_sum',␣
 ↪'segment_osrm_distance_sum',
            'segment_osrm_time_sum', 'hour_taken']

num_col
```

[48]:
```
['start_scan_to_end_scan',
 'actual_distance_to_destination',
 'actual_time',
 'osrm_time',
 'osrm_distance',
 'segment_actual_time_sum',
 'segment_osrm_distance_sum',
 'segment_osrm_time_sum',
 'hour_taken']
```

2.3 Find outliers in numerical variable

[49]:
```
trip[num_col].boxplot(rot=25, figsize=(30,10))
```

[49]:
```
<Axes: >
```



as we can observe, every numerical column has outliers

2.4 Handle the outliers using IQR method

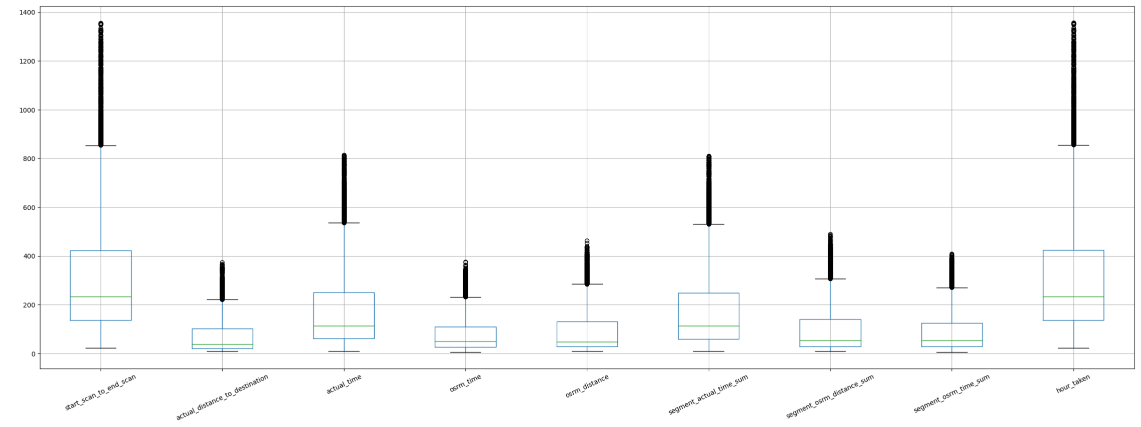
```
[50]: Q1 = trip[num_col].quantile(0.25) #Calculates the first quartile (25th␣
       ↪percentile)
      Q3 = trip[num_col].quantile(0.75)  # Calculates the third quartile (75th␣
       ↪percentile)
      IQR = Q3 - Q1 # Computes the Interquartile Range

      # Filtering out the Outliers, keeping the rowa that are not outliers from trip␣
       ↪dataframe

      trip = trip[~((trip[num_col] < (Q1 - 1.5 * IQR)) | (trip[num_col] > (Q3 + 1.5 *␣
       ↪IQR))).any(axis=1)]
      trip = trip.reset_index(drop=True)

      trip.head( 3)
```

```
[50]:        data         trip_creation_time  \
      0  training 2018-09-12 00:00:22.886430
      1  training 2018-09-12 00:01:00.113710
      2  training 2018-09-12 00:02:09.740725

                                       route_schedule_uuid route_type  \
      0  thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…    Carting
      1  thanos::sroute:f0176492-a679-4597-8332-bbd1c7f…    Carting
      2  thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134…        FTL

                     trip_uuid source_center                         source_name  \
      0  trip-153671042288605164  IND561203AAB  doddablpur_chikadpp_d (karnataka)
      1  trip-153671046011330457  IND400072AAB           mumbai hub (maharashtra)
```

```
2  trip-153671052974046625  IND583101AAA             bellary_dc (karnataka)

  destination_center                    destination_name  \
0       IND561203AAB  doddablpur_chikadpp_d (karnataka)
1       IND401104AAA     mumbai_mirard_ip (maharashtra)
2       IND583119AAA      sandur_wrdn1dpp_d (karnataka)

   start_scan_to_end_scan  …  source_state  source_city  source_place  \
0                   180.0  …     karnataka   doddablpur      chikadpp
1                   100.0  …   maharashtra  mumbai hub        mumbai
2                   717.0  …     karnataka      bellary       bellary

   source_code  trip_year  trip_month  trip_hour  trip_day trip_week  \
0            d       2018           9          0        12        37
1         none       2018           9          0        12        37
2         none       2018           9          0        12        37

  trip_dayofweek
0              2
1              2
2              2

[3 rows x 32 columns]
```

[51]:
```python
# getting the chart after filtering the trip dataframe
trip[num_col].boxplot(rot=25, figsize=(30,10))
```

[51]:
```
<Axes: >
```



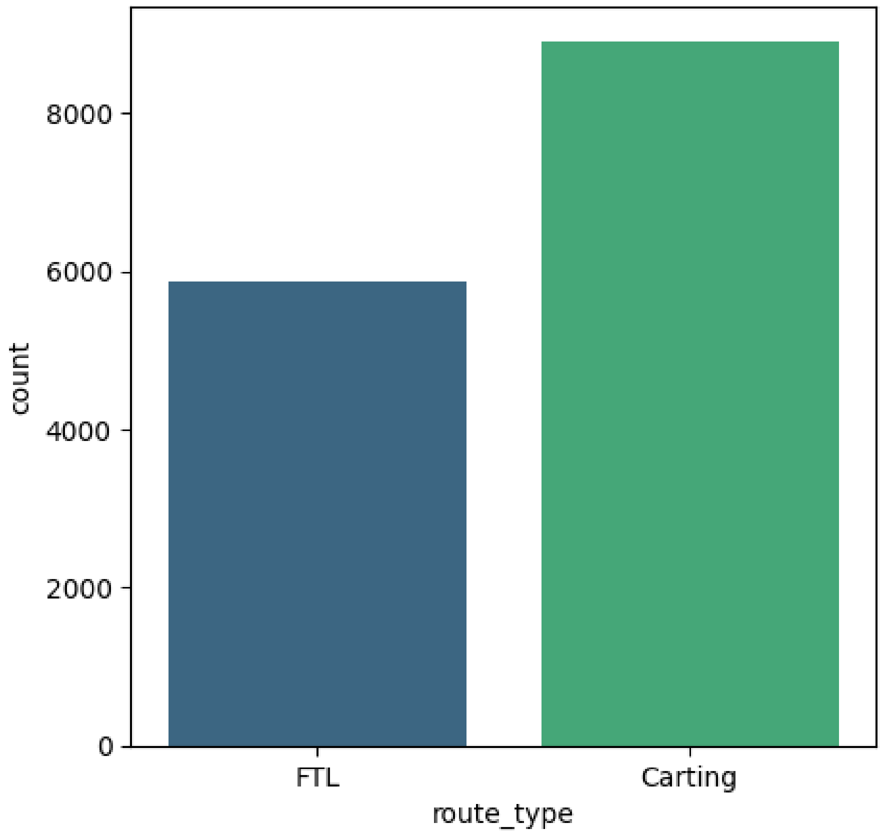
2.5 Categorical Variables

[89]:
```
# getting the count of two types of route types
trip['route_type'].value_counts()
```

[89]:
```
route_type
Carting     8906
FTL         5881
Name: count, dtype: int64
```

[91]:
```
# visualization of above chart
plt.figure(figsize=(5,5))
sns.countplot(data = trip, x = 'route_type',palette='viridis')
```

[91]:
```
<Axes: xlabel='route_type', ylabel='count'>
```



[56]:
```
#mapping the 2 trpes of routes
trip['route_type'] = trip['route_type'].map({'FTL':0, 'Carting':1})
```

[58]:
```
trip
```

```
[58]:          data         trip_creation_time  \
0      training 2018-09-12 00:00:22.886430
1      training 2018-09-12 00:01:00.113710
2      training 2018-09-12 00:02:09.740725
3      training 2018-09-12 00:02:34.161600
4      training 2018-09-12 00:04:22.011653
…         …                          …
12718      test 2018-10-03 23:55:56.258533
12719      test 2018-10-03 23:57:23.863155
12720      test 2018-10-03 23:57:44.429324
12721      test 2018-10-03 23:59:14.390954
12722      test 2018-10-03 23:59:42.701692

                                    route_schedule_uuid  route_type  \
0      thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0…        NaN
1      thanos::sroute:f0176492-a679-4597-8332-bbd1c7f…        NaN
2      thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134…        NaN
3      thanos::sroute:9bf03170-d0a2-4a3f-aa4d-9aaab3d…        NaN
4      thanos::sroute:a97698cc-846e-41a7-916b-88b1741…        NaN
…                                                   …          …
12718  thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14…        NaN
12719  thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769…        NaN
12720  thanos::sroute:5609c268-e436-4e0a-8180-3db4a74…        NaN
12721  thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a…        NaN
12722  thanos::sroute:412fea14-6d1f-4222-8a5f-a517042…        NaN

                     trip_uuid source_center  \
0      trip-153671042288605164  IND561203AAB
1      trip-153671046011330457  IND400072AAB
2      trip-153671052974046625  IND583101AAA
3      trip-153671055416136166  IND600056AAA
4      trip-153671066201138152  IND600044AAD
…                          …            …
12718  trip-153861095625827784  IND160002AAC
12719  trip-153861104386292051  IND121004AAB
12720  trip-153861106442901555  IND208006AAA
12721  trip-153861115439069069  IND627005AAA
12722  trip-153861118270144424  IND583119AAA

                                source_name destination_center  \
0        doddablpur_chikadpp_d (karnataka)       IND561203AAB
1                  mumbai hub (maharashtra)       IND401104AAA
2                    bellary_dc (karnataka)       IND583119AAA
3         chennai_poonamallee (tamil nadu)       IND600056AAA
4          chennai_chrompet_dpc (tamil nadu)       IND600048AAA
…                                       …              …
12718       chandigarh_mehmdpur_h (punjab)       IND160002AAC
```

```
12719        fbd_balabhgarh_dpc (haryana)       IND121004AAA
12720   kanpur_govndngr_dc (uttar pradesh)       IND208006AAA
12721  tirunelveli_vdkkusrt_i (tamil nadu)       IND628204AAA
12722        sandur_wrdn1dpp_d (karnataka)       IND583119AAA

                          destination_name  start_scan_to_end_scan  …  \
0       doddablpur_chikadpp_d (karnataka)                   180.0  …
1          mumbai_mirard_ip (maharashtra)                   100.0  …
2           sandur_wrdn1dpp_d (karnataka)                   717.0  …
3         chennai_poonamallee (tamil nadu)                  189.0  …
4         chennai_vandalur_dc (tamil nadu)                   98.0  …
…                                      …                     …  …
12718      chandigarh_mehmdpur_h (punjab)                   257.0  …
12719      faridabad_blbgarh_dc (haryana)                    60.0  …
12720  kanpur_govndngr_dc (uttar pradesh)                   421.0  …
12721  tirchchndr_shnmgprm_d (tamil nadu)                   347.0  …
12722       sandur_wrdn1dpp_d (karnataka)                   353.0  …

        source_state  source_city  source_place  source_code  trip_year  \
0          karnataka   doddablpur      chikadpp            d       2018
1        maharashtra  mumbai hub         mumbai         none       2018
2          karnataka      bellary       bellary         none       2018
3         tamil nadu      chennai       chennai         none       2018
4         tamil nadu      chennai      chrompet          dpc       2018
…                …           …            …            …         …
12718         punjab   chandigarh      mehmdpur            h       2018
12719        haryana          fbd    balabhgarh          dpc       2018
12720  uttar pradesh       kanpur      govndngr           dc       2018
12721     tamil nadu  tirunelveli      vdkkusrt            i       2018
12722      karnataka       sandur      wrdn1dpp            d       2018

       trip_month  trip_hour  trip_day trip_week trip_dayofweek
0               9          0        12        37              2
1               9          0        12        37              2
2               9          0        12        37              2
3               9          0        12        37              2
4               9          0        12        37              2
…             …          …        …        …              …
12718          10         23         3        40              2
12719          10         23         3        40              2
12720          10         23         3        40              2
12721          10         23         3        40              2
12722          10         23         3        40              2

[12723 rows x 32 columns]
```

2.5.1 Standardize the numerical features using StandardScaler

[61]:
```
from sklearn.preprocessing import StandardScaler

# standardize features and calculates the mean and standard deviation for each␣
↪of these columns.
scaler = StandardScaler()
scaler.fit(trip[num_col])
```

[61]:
```
StandardScaler()
```

[63]:
```
#transform the specified numerical columns, making them have a mean of 0 and a␣
↪standard deviation of 1
trip[num_col] = scaler.transform(trip[num_col])
trip[num_col]
```

[63]:
```
       start_scan_to_end_scan  actual_distance_to_destination  actual_time  \
0                   -1.255068                       -1.003307    -1.123469
1                   -1.256293                       -1.014092    -1.126828
2                   -1.246845                       -0.992860    -1.115552
3                   -1.254930                       -1.012663    -1.126748
4                   -1.256323                       -1.015647    -1.128227
…                         …                              …            …
12718               -1.253889                       -1.006277    -1.125868
12719               -1.256905                       -1.014412    -1.128347
12720               -1.251377                       -1.009950    -1.117911
12721               -1.252510                       -0.991459    -1.118631
12722               -1.252419                       -1.004675    -1.118191

       osrm_time  osrm_distance  segment_actual_time_sum  \
0      -1.086461      -1.025065                -1.120464
1      -1.096592      -1.033222                -1.123787
2      -1.077095      -1.017376                -1.112401
3      -1.095063      -1.032177                -1.123747
4      -1.096974      -1.034177                -1.125205
…          …              …                        …
12718  -1.087608      -1.026517                -1.122855
12719  -1.097165      -1.033670                -1.125327
12720  -1.090284      -1.028332                -1.114791
12721  -1.065245      -1.014344                -1.115723
12722  -1.086461      -1.025630                -1.115075

       segment_osrm_distance_sum  segment_osrm_time_sum  hour_taken
0                      -1.034821              -1.082525   -1.256735
1                      -1.041975              -1.090258   -1.257974
2                      -1.027858              -1.074634   -1.248537
3                      -1.041064              -1.089154   -1.256599
```

```
4                      -1.042849           -1.090732   -1.258012
…                          …                   …           …
12718                  -1.036972           -1.082999   -1.255568
12719                  -1.042397           -1.091047   -1.258583
12720                  -1.032519           -1.078896   -1.253062
12721                  -1.019322           -1.057906   -1.254186
12722                  -1.035223           -1.082210   -1.254096

[12723 rows x 9 columns]
```

[65]:
```
#get a statistical summary of a specific numerical columns in the DataFrame trip
trip[num_col].describe()
```

[65]:
```
       start_scan_to_end_scan  actual_distance_to_destination   actual_time  \
count            12723.000000                    12723.000000  12723.000000
mean                -1.252921                       -1.003475     -1.122091
std                  0.003913                        0.013876      0.006324
min                 -1.257472                       -1.015665     -1.128827
25%                 -1.255742                       -1.013279     -1.126748
50%                 -1.254256                       -1.009981     -1.124629
75%                 -1.251347                       -0.997823     -1.119151
max                 -1.237075                       -0.945496     -1.096599

          osrm_time  osrm_distance  segment_actual_time_sum  \
count  12723.000000   12723.000000             12723.000000
mean      -1.084466      -1.024240                -1.119054
std        0.013826       0.011166                 0.006366
min       -1.098312      -1.034545                -1.125813
25%       -1.094298      -1.032142                -1.123747
50%       -1.089902      -1.029640                -1.121599
75%       -1.078624      -1.019305                -1.116129
max       -1.027590      -0.977896                -1.093356

       segment_osrm_distance_sum  segment_osrm_time_sum    hour_taken
count               12723.000000           12723.000000  12723.000000
mean                   -1.033288              -1.079226     -1.254606
std                     0.010547               0.012563      0.003908
min                    -1.043177              -1.091837     -1.259151
25%                    -1.040912              -1.088365     -1.257424
50%                    -1.038169              -1.084419     -1.255942
75%                    -1.028554              -1.073056     -1.253034
max                    -0.989729              -1.028394     -1.238776
```

There is a significant difference between OSRM and actual parameters.

3 Recomendation

1. There is a significant difference between OSRM and actual parameters.
2. We need to check information fed to routing engine for trip planning.
3. North, South and West Zones have significant numbers of orders.
4. we need to increasing our presence in Central, Eastern and North-Eastern zone. As we have small presence in these area.
5. we have maximum number of orders in Mahrashtra followed by Karnataka.
6. we need to prepare for resources on ground level in these states on festivels

[ ]:

```
```