delhivery-data-1

May 3, 2024

0.0.1 IMPORTS

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

1 1. Define Problem Statement and perform Exploratory Data Analysis (10 points)

Definition of problem (as per given problem statement with additional views) Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary. Visual Analysis (distribution plots of all the continuous variable(s), boxplots of all the categorical variables) Insights based on EDA Comments on range of attributes, outliers of various attributes Comments on the distribution of the variables and relationship between them Comments for each univariate and bivariate plot

1.0.1 Problem Statement

Answer: Enhance Delhivery's operational efficiency, forecast accuracy, and profitability through advanced data analytics.

1.0.2 Data Loading and Initial Exploration

[2]:
```
# Load the dataset
df = pd.read_csv('/content/drive/MyDrive/delhivery/delhivery_data.csv')
```

[3]:
```
df.head()
```

[3]:
```
       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
3  training  2018-09-20 02:35:36.476840
4  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
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4  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)
3  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4  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)
3       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4       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
3  2018-09-20 03:21:32.418600  …         2018-09-20 03:39:57
4  2018-09-20 03:21:32.418600  …         2018-09-20 03:33:55

   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
3                       36.118028         62.0       40.0       45.5620
4                       39.386040         68.0       44.0       54.2181

     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
3  1.550000                 21.0               12.0                13.0224
4  1.545455                  6.0                5.0                 3.9153

   segment_factor
0        1.272727
1        1.111111
2        2.285714
3        1.750000
4        1.200000
```

```
[5 rows x 24 columns]
```

[4]:
```
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
```

1.1 Basic data cleaning and exploration:

1. Handle missing values in the data.
2. Analyze the structure of the data.
3. Try merging the rows using the hint mentioned above.

1.1 Handle missing values in the data.

[5]:
```
df.isnull().sum()
```

[5]:
```
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
```

Only source_name and destination_name has got null values. There is absolutely no missing values in any numeric column.

[6]:
```
df_dropped = df.dropna()
columns_to_fill = ['source_name','destination_name']
for column in columns_to_fill:
  df_dropped[column].fillna('Unknown', inplace = True)
df_dropped.isna().sum()
```

```
<ipython-input-6-1a7ace2663c8>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[column].fillna('Unknown', inplace = True)
```

[6]:
```
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
```

[7]:
```
time_columns = ['od_start_time', 'od_end_time', 'cutoff_timestamp']
duration_columns = ['start_scan_to_end_scan', 'actual_time', 'osrm_time',␣
 ↪'segment_actual_time', 'segment_osrm_time']
distance_columns = ['actual_distance_to_destination', 'osrm_distance',␣
 ↪'segment_osrm_distance']
boolean_columns = ['is_cutoff']
factor_columns = ['cutoff_factor', 'factor', 'segment_factor']
for col in time_columns:
  df_dropped[col].fillna(method = 'ffill', inplace = True)
for col in duration_columns + distance_columns + factor_columns :
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
for col in boolean_columns:
    df_dropped[col].fillna(df_dropped[col].mode()[0], inplace=True)
df_dropped
```

```
<ipython-input-7-dbb4efc4521f>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(method = 'ffill', inplace = True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].median(), inplace=True)
<ipython-input-7-dbb4efc4521f>:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped[col].fillna(df_dropped[col].mode()[0], inplace=True)
```

[7]:

	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			
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting			
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting			
…	…	…	…	…			
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…	Carting			
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…	Carting			
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…	Carting			
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…	Carting			
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…	Carting			

```
2       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
…                          …           …                          …
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)

       destination_center              destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
…                      …                           …
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                    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
3       2018-09-20 03:21:32.418600  …         2018-09-20 03:39:57
4       2018-09-20 03:21:32.418600  …         2018-09-20 03:33:55
…                              …  …                          …
144862  2018-09-20 16:24:28.436231  …         2018-09-20 21:57:20
144863  2018-09-20 16:24:28.436231  …         2018-09-20 21:31:18
144864  2018-09-20 16:24:28.436231  …         2018-09-20 21:11:18
144865  2018-09-20 16:24:28.436231  …         2018-09-20 20:53:19
144866  2018-09-20 16:24:28.436231  …  2018-09-20 16:24:28.436231

        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
3                            36.118028         62.0       40.0       45.5620
4                            39.386040         68.0       44.0       54.2181
…                                  …            …          …            …
144862                       45.258278         94.0       60.0       67.9280
144863                       54.092531        120.0       76.0       85.6829
144864                       66.163591        140.0       88.0       97.0933
144865                       73.680667        158.0       98.0      111.2709
```

```
144866                      70.039010        426.0        95.0       88.7319

          factor  segment_actual_time  segment_osrm_time  \
0       1.272727                 14.0               11.0
1       1.200000                 10.0                9.0
2       1.428571                 16.0                7.0
3       1.550000                 21.0               12.0
4       1.545455                  6.0                5.0
…          …                      …                  …
144862  1.566667                 12.0               12.0
144863  1.578947                 26.0               21.0
144864  1.590909                 20.0               34.0
144865  1.612245                 17.0               27.0
144866  4.484211                268.0                9.0

        segment_osrm_distance  segment_factor
0                     11.9653        1.272727
1                      9.7590        1.111111
2                     10.8152        2.285714
3                     13.0224        1.750000
4                      3.9153        1.200000
…                         …              …
144862                 8.1858        1.000000
144863                17.3725        1.238095
144864                20.7053        0.588235
144865                18.8885        0.629630
144866                 8.8088       29.777778

[144316 rows x 24 columns]
```

1.2 Analyze the structure of the data.

[8]:
```
df_dropped.head()
```

[8]:
```
       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
3  training  2018-09-20 02:35:36.476840
4  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
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4  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)
3  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4  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)
3       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4       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
3  2018-09-20 03:21:32.418600  …         2018-09-20 03:39:57
4  2018-09-20 03:21:32.418600  …         2018-09-20 03:33:55

   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
3                       36.118028         62.0       40.0       45.5620
4                       39.386040         68.0       44.0       54.2181

     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
3  1.550000                 21.0               12.0                13.0224
4  1.545455                  6.0                5.0                 3.9153

   segment_factor
0        1.272727
1        1.111111
2        2.285714
3        1.750000
4        1.200000

[5 rows x 24 columns]
```

[9]:
```
df_dropped.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 144316 entries, 0 to 144866
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   data                            144316 non-null  object
 1   trip_creation_time              144316 non-null  object
 2   route_schedule_uuid             144316 non-null  object
 3   route_type                      144316 non-null  object
 4   trip_uuid                       144316 non-null  object
 5   source_center                   144316 non-null  object
 6   source_name                     144316 non-null  object
 7   destination_center              144316 non-null  object
 8   destination_name                144316 non-null  object
 9   od_start_time                   144316 non-null  object
 10  od_end_time                     144316 non-null  object
 11  start_scan_to_end_scan          144316 non-null  float64
 12  is_cutoff                       144316 non-null  bool
 13  cutoff_factor                   144316 non-null  int64
 14  cutoff_timestamp                144316 non-null  object
 15  actual_distance_to_destination  144316 non-null  float64
 16  actual_time                     144316 non-null  float64
 17  osrm_time                       144316 non-null  float64
 18  osrm_distance                   144316 non-null  float64
 19  factor                          144316 non-null  float64
 20  segment_actual_time             144316 non-null  float64
 21  segment_osrm_time               144316 non-null  float64
 22  segment_osrm_distance           144316 non-null  float64
 23  segment_factor                  144316 non-null  float64
dtypes: bool(1), float64(10), int64(1), object(12)
memory usage: 26.6+ MB
```

[10]:
```
df_dropped.describe()
```

[10]:

	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	\
count	144316.000000	144316.000000	144316.000000	
mean	963.697698	233.561345	234.708498	
std	1038.082976	345.245823	345.480571	
min	20.000000	9.000000	9.000045	
25%	161.000000	22.000000	23.352027	
50%	451.000000	66.000000	66.135322	
75%	1645.000000	286.000000	286.919294	
max	7898.000000	1927.000000	1927.447705	

	actual_time	osrm_time	osrm_distance	factor	\
count	144316.000000	144316.000000	144316.000000	144316.000000	
mean	417.996237	214.437055	285.549785	2.120178	

```
std       598.940065     308.448543     421.717826       1.717065
min         9.000000       6.000000       9.008200       0.144000
25%        51.000000      27.000000      29.896250       1.604545
50%       132.000000      64.000000      78.624400       1.857143
75%       516.000000     259.000000     346.305400       2.212280
max      4532.000000    1686.000000    2326.199100      77.387097

       segment_actual_time  segment_osrm_time  segment_osrm_distance  \
count        144316.000000      144316.000000          144316.000000
mean             36.175379          18.495697              22.818993
std              53.524298          14.774008              17.866367
min            -244.000000           0.000000               0.000000
25%              20.000000          11.000000              12.053975
50%              28.000000          17.000000              23.508300
75%              40.000000          22.000000              27.813325
max            3051.000000        1611.000000            2191.403700

       segment_factor
count   144316.000000
mean         2.218707
std          4.854804
min        -23.444444
25%          1.347826
50%          1.684211
75%          2.250000
max        574.250000
```

Statistical summary for categorical columns

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

[11]:
```
            data          trip_creation_time  \
count     144316                      144316
unique         2                       14787
top     training  2018-10-01 05:04:55.268931
freq       104632                         101

                                   route_schedule_uuid route_type  \
count                                           144316     144316
unique                                            1497          2
top     thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069f…        FTL
freq                                              1812      99132

                     trip_uuid source_center                    source_name  \
count                   144316        144316                         144316
unique                   14787          1496                           1496
top     trip-153837029526866991  IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
```

```
freq                        101         23267                           23267

       destination_center              destination_name  \
count              144316                        144316
unique               1466                          1466
top           IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
freq                15192                         15192

                     od_start_time                 od_end_time  \
count                       144316                      144316
unique                       26223                       26223
top     2018-09-21 18:37:09.322207  2018-09-24 09:59:15.691618
freq                            81                          81

           cutoff_timestamp
count                144316
unique                92894
top     2018-09-24 05:19:20
freq                     39
```

data types

[12]:
```
df_dropped.dtypes
```

[12]:
```
data                            object
trip_creation_time              object
route_schedule_uuid             object
route_type                      object
trip_uuid                       object
source_center                   object
source_name                     object
destination_center              object
destination_name                object
od_start_time                   object
od_end_time                     object
start_scan_to_end_scan         float64
is_cutoff                         bool
cutoff_factor                    int64
cutoff_timestamp                object
actual_distance_to_destination  float64
actual_time                     float64
osrm_time                       float64
osrm_distance                   float64
factor                          float64
segment_actual_time             float64
segment_osrm_time               float64
segment_osrm_distance           float64
```

```
segment_factor                    float64
dtype: object
```

Unique values

[13]:
```
for column in df_dropped.columns:
  print(f"{column} has {df_dropped[column].nunique()} unique values")
```

```
data has 2 unique values
trip_creation_time has 14787 unique values
route_schedule_uuid has 1497 unique values
route_type has 2 unique values
trip_uuid has 14787 unique values
source_center has 1496 unique values
source_name has 1496 unique values
destination_center has 1466 unique values
destination_name has 1466 unique values
od_start_time has 26223 unique values
od_end_time has 26223 unique values
start_scan_to_end_scan has 1914 unique values
is_cutoff has 2 unique values
cutoff_factor has 501 unique values
cutoff_timestamp has 92894 unique values
actual_distance_to_destination has 143965 unique values
actual_time has 3182 unique values
osrm_time has 1531 unique values
osrm_distance has 137544 unique values
factor has 45588 unique values
segment_actual_time has 746 unique values
segment_osrm_time has 214 unique values
segment_osrm_distance has 113497 unique values
segment_factor has 5663 unique values
```

Value Counts for Categorical Data

[14]:
```
for column in df_dropped.select_dtypes(include=['object']).columns:
    print(f"Value counts for {column}:")
    print(df_dropped[column].value_counts())
    print("\n")
```

```
Value counts for data:
data
training    104632
test         39684
Name: count, dtype: int64


Value counts for trip_creation_time:
```

```
trip_creation_time
2018-10-01 05:04:55.268931    101
2018-09-17 04:43:09.467353    101
2018-09-28 05:23:15.359220    101
2018-10-03 05:04:14.929361    101
2018-09-27 04:47:19.425867    101
                              …
2018-09-17 01:26:45.903862      1
2018-09-19 06:09:03.938546      1
2018-10-03 18:45:56.568239      1
2018-10-03 23:47:01.484016      1
2018-10-03 22:22:54.493221      1
Name: count, Length: 14787, dtype: int64


Value counts for route_schedule_uuid:
route_schedule_uuid
thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069fbcea9    1812
thanos::sroute:0456b740-1dad-4929-bbe0-87d8843f5a10    1608
thanos::sroute:dca6268f-741a-4d1a-b1b0-aab13095a366    1605
thanos::sroute:a1b25549-1e77-498f-8538-00292e5bd5a2    1285
thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e5720d    1280
                                                       …
thanos::sroute:c6403e98-1ffa-49ed-bc55-5adcdc4ad39d       1
thanos::sroute:889b9cf5-da6a-48ce-b3bd-6983c8090164       1
thanos::sroute:d563d17e-2123-40a4-9eec-40018966caba       1
thanos::sroute:afb04bf4-7734-4526-8d66-63ed50bef88b       1
thanos::sroute:02447388-0297-4e63-9a71-3589741bd807       1
Name: count, Length: 1497, dtype: int64


Value counts for route_type:
route_type
FTL        99132
Carting    45184
Name: count, dtype: int64


Value counts for trip_uuid:
trip_uuid
trip-153837029526866991    101
trip-153715938946690081    101
trip-153811219535896559    101
trip-153854305492910872    101
trip-153802363942560700    101
                           …
trip-153714761197530398      1
trip-153733734393817090      1
```

```
trip-153859235656799725      1
trip-153861042148375909      1
trip-153860537449292853      1
Name: count, Length: 14787, dtype: int64


Value counts for source_center:
source_center
IND000000ACB    23267
IND562132AAA     9975
IND421302AAG     9088
IND411033AAA     4061
IND501359AAE     3340
                …
IND741121AAA        1
IND493445AAB        1
IND621112AAA        1
IND733202AAC        1
IND335501AAA        1
Name: count, Length: 1496, dtype: int64


Value counts for source_name:
source_name
Gurgaon_Bilaspur_HB (Haryana)            23267
Bangalore_Nelmngla_H (Karnataka)          9975
Bhiwandi_Mankoli_HB (Maharashtra)         9088
Pune_Tathawde_H (Maharashtra)             4061
Hyderabad_Shamshbd_H (Telangana)          3340
                                         …
Badkulla_Central_DPP_1 (West Bengal)         1
Mahasamund_RajpurRD_D (Chhattisgarh)         1
Tiruchi_Samyaprm_D (Tamil Nadu)              1
Islampure_Central_DPP_2 (West Bengal)        1
Bhadra_GMndiDPP_D (Rajasthan)                1
Name: count, Length: 1496, dtype: int64


Value counts for destination_center:
destination_center
IND000000ACB    15192
IND562132AAA    11019
IND421302AAG     5492
IND501359AAE     5142
IND712311AAA     4892
                …
IND490023AAA        1
IND221401AAA        1
```

```
IND520011AAA        1
IND396210AAA        1
IND761020AAA        1
Name: count, Length: 1466, dtype: int64


Value counts for destination_name:
destination_name
Gurgaon_Bilaspur_HB (Haryana)        15192
Bangalore_Nelmngla_H (Karnataka)     11019
Bhiwandi_Mankoli_HB (Maharashtra)     5492
Hyderabad_Shamshbd_H (Telangana)      5142
Kolkata_Dankuni_HB (West Bengal)      4892
                                     …
Durg_Bhilai_DC (Chhattisgarh)            1
Bhadohi_Rajpura_D (Uttar Pradesh)        1
Vijayawada (Andhra Pradesh)              1
Daman_DC (Daman & Diu)                   1
Berhampur_Chatrpr_DC (Orissa)            1
Name: count, Length: 1466, dtype: int64


Value counts for od_start_time:
od_start_time
2018-09-21 18:37:09.322207    81
2018-09-27 06:12:46.137400    79
2018-09-14 07:13:24.396869    79
2018-10-03 04:55:30.039225    79
2018-09-21 06:51:50.532257    79
                              ..
2018-09-12 20:24:29.188095     1
2018-09-14 02:03:46.452392     1
2018-09-18 06:19:09.183297     1
2018-10-02 21:47:34.760258     1
2018-09-23 22:44:33.236923     1
Name: count, Length: 26223, dtype: int64


Value counts for od_end_time:
od_end_time
2018-09-24 09:59:15.691618    81
2018-09-29 12:11:18.125562    79
2018-09-16 17:00:03.263746    79
2018-10-05 11:15:01.115906    79
2018-09-23 13:03:50.216955    79
                              ..
2018-09-13 03:42:55.423049     1
2018-09-14 03:25:53.825948     1
```

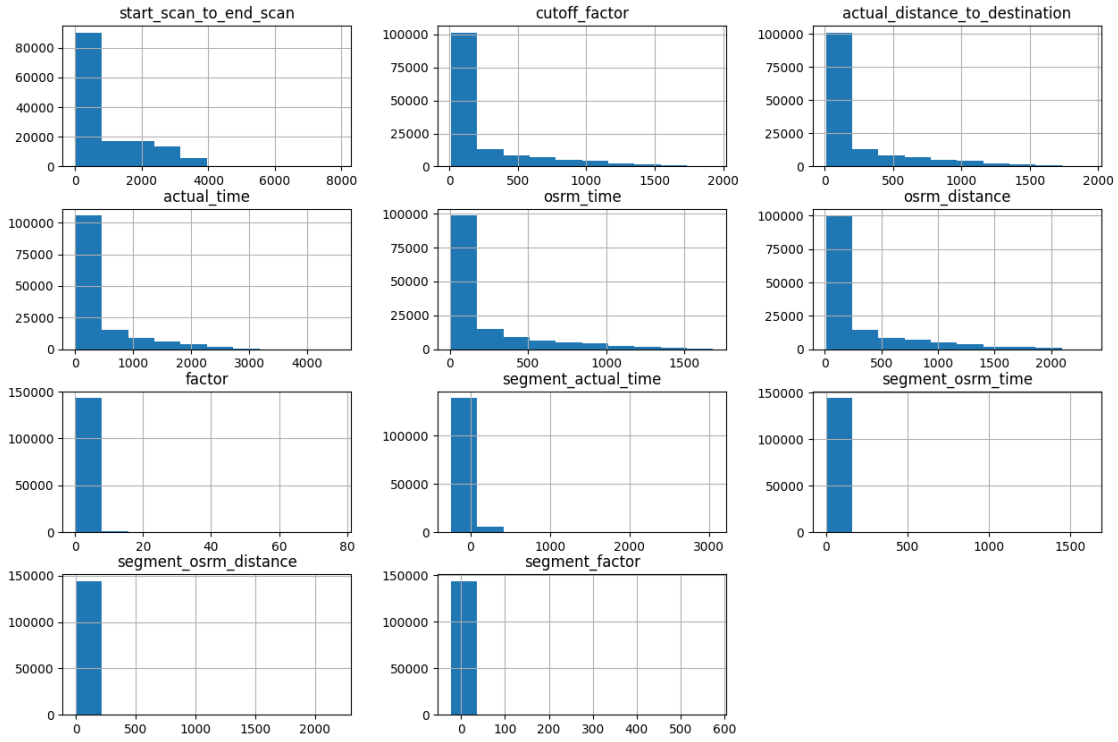
```
2018-09-18 07:40:56.344901     1
2018-10-03 01:55:45.736730     1
2018-09-24 15:17:24.668332     1
Name: count, Length: 26223, dtype: int64


Value counts for cutoff_timestamp:
cutoff_timestamp
2018-09-24 05:19:20           39
2018-09-24 05:19:21           33
2018-09-14 05:29:26           19
2018-09-24 07:21:24           18
2018-09-14 02:35:24           17
                              ..
2018-09-30 23:20:12            1
2018-09-30 22:48:12            1
2018-09-27 01:21:55            1
2018-09-27 01:15:55            1
2018-09-20 16:24:28.436231     1
Name: count, Length: 92894, dtype: int64
```

[14]:

2 Visualize the data

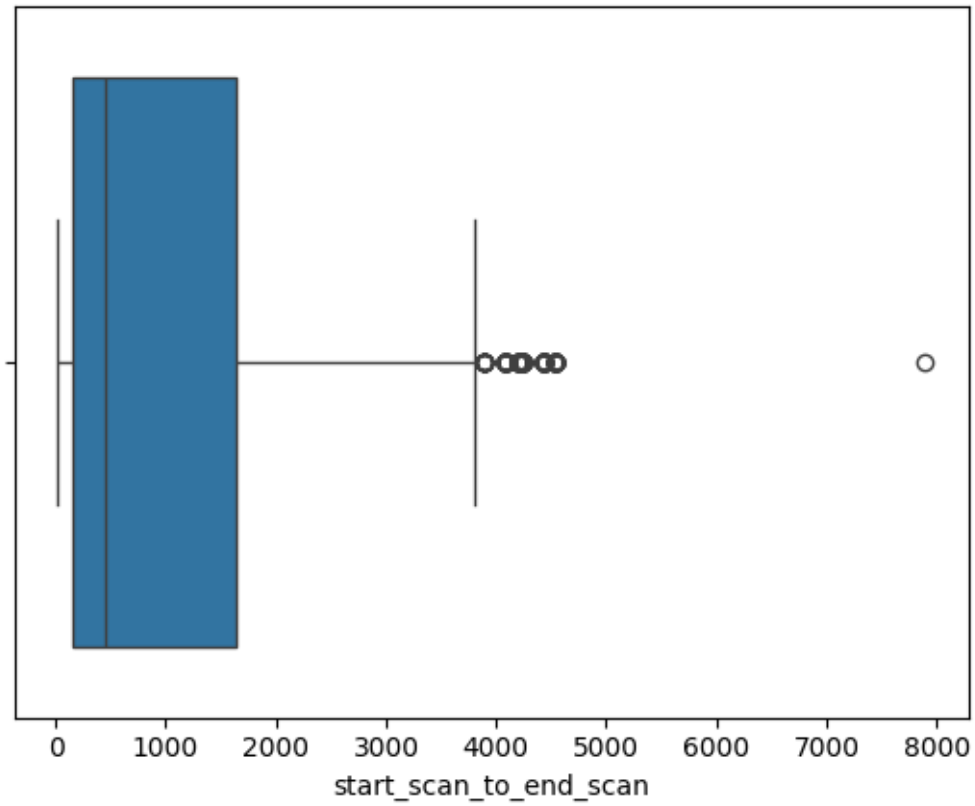
(histogram for numerical data)

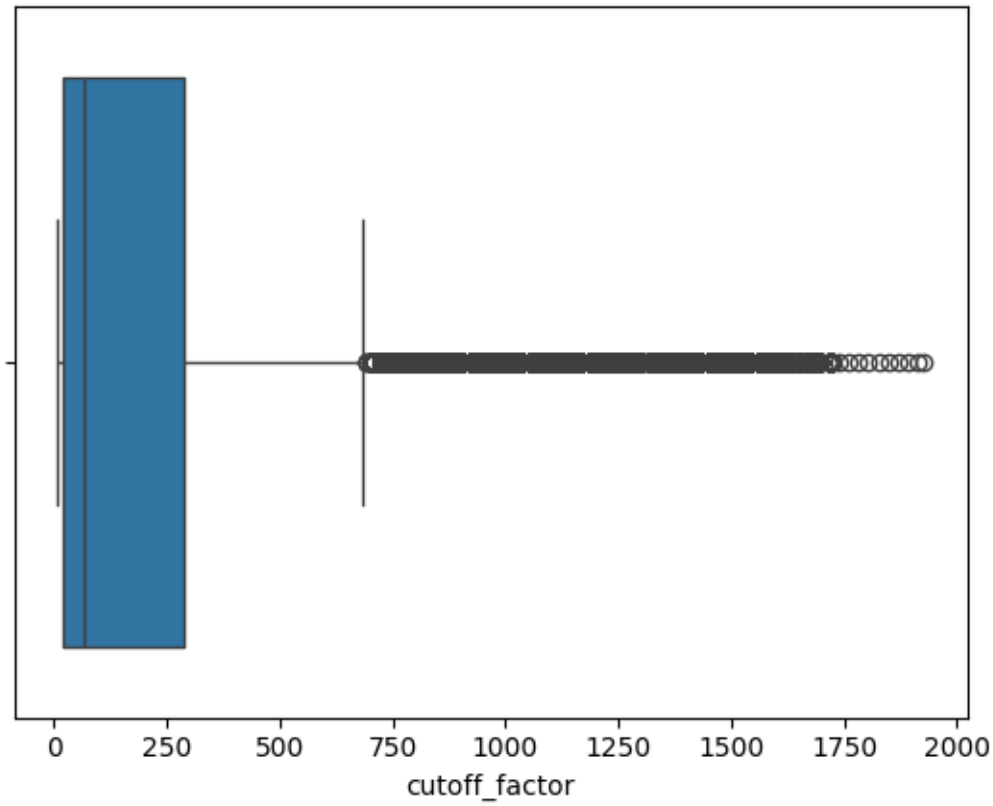
[15]:
```
import matplotlib.pyplot as plt
import seaborn as sns
df_dropped.hist(figsize = (15,10))
plt.show()
```

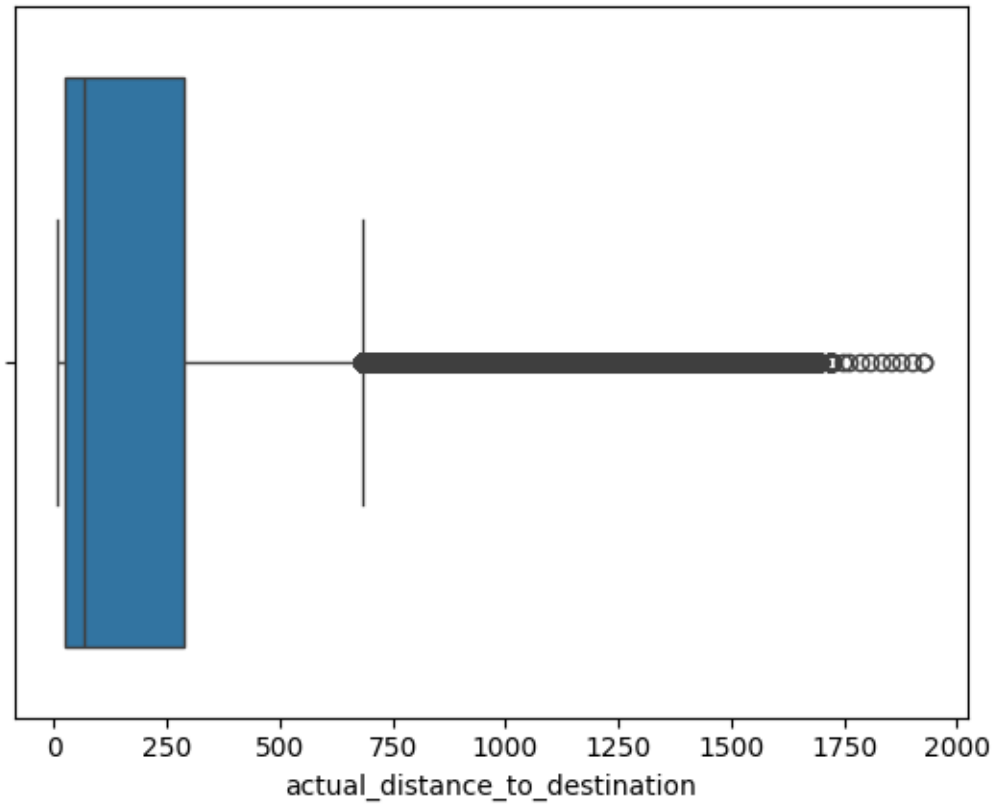


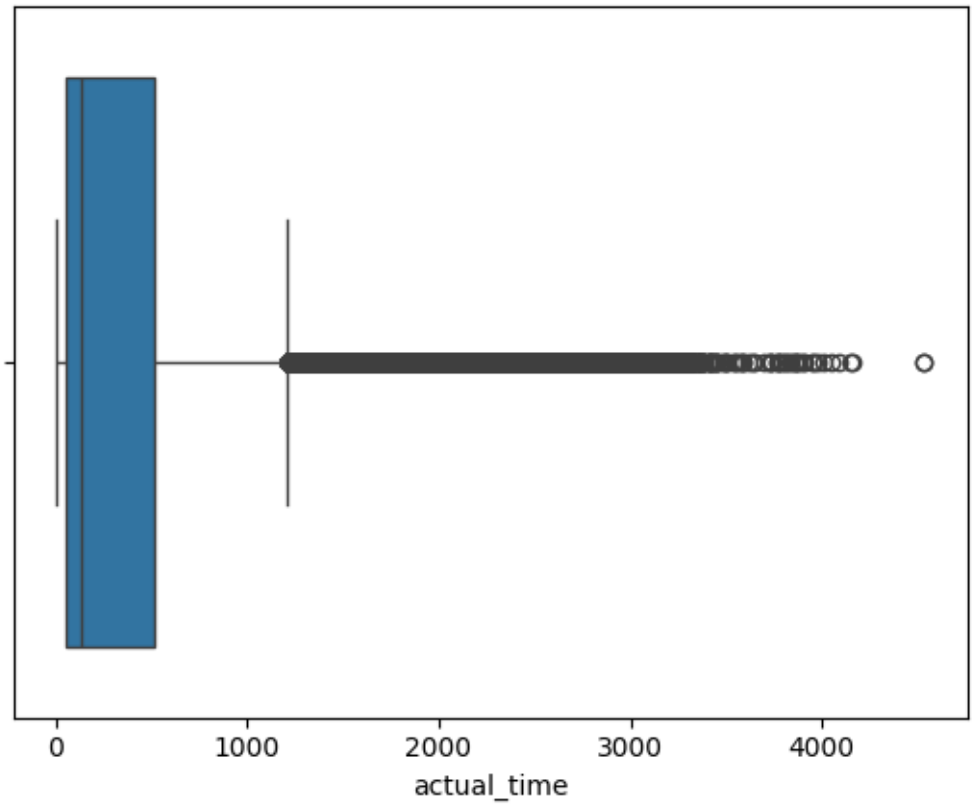
For Outliers(Boxplot)

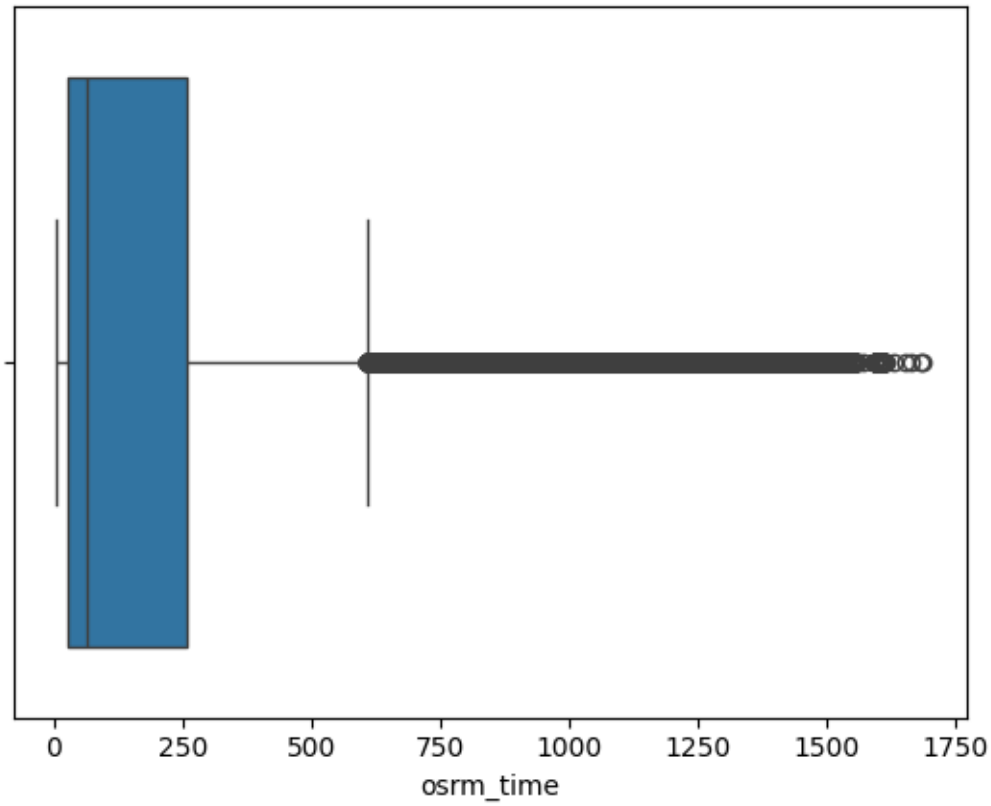
[16]:
```
for column in df_dropped.select_dtypes(include=['number']).columns:
    sns.boxplot(x=df_dropped[column])
    plt.show()
```

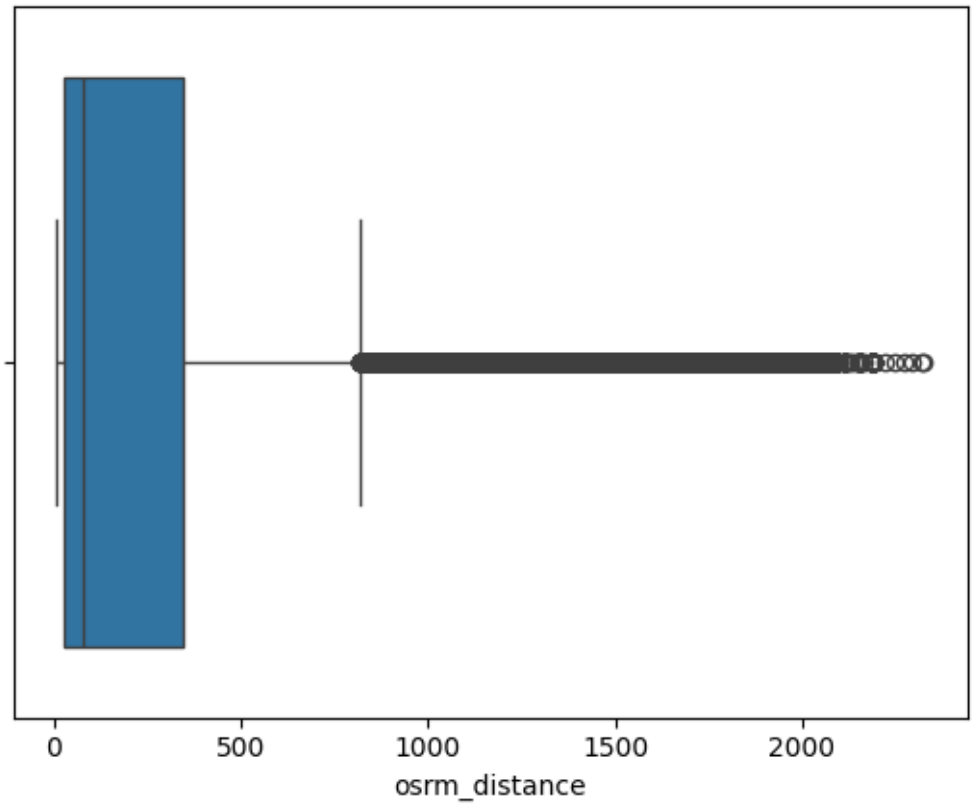

0
1000
2000
3000
4000
5000
6000
7000
8000
start_scan_to_end_scan

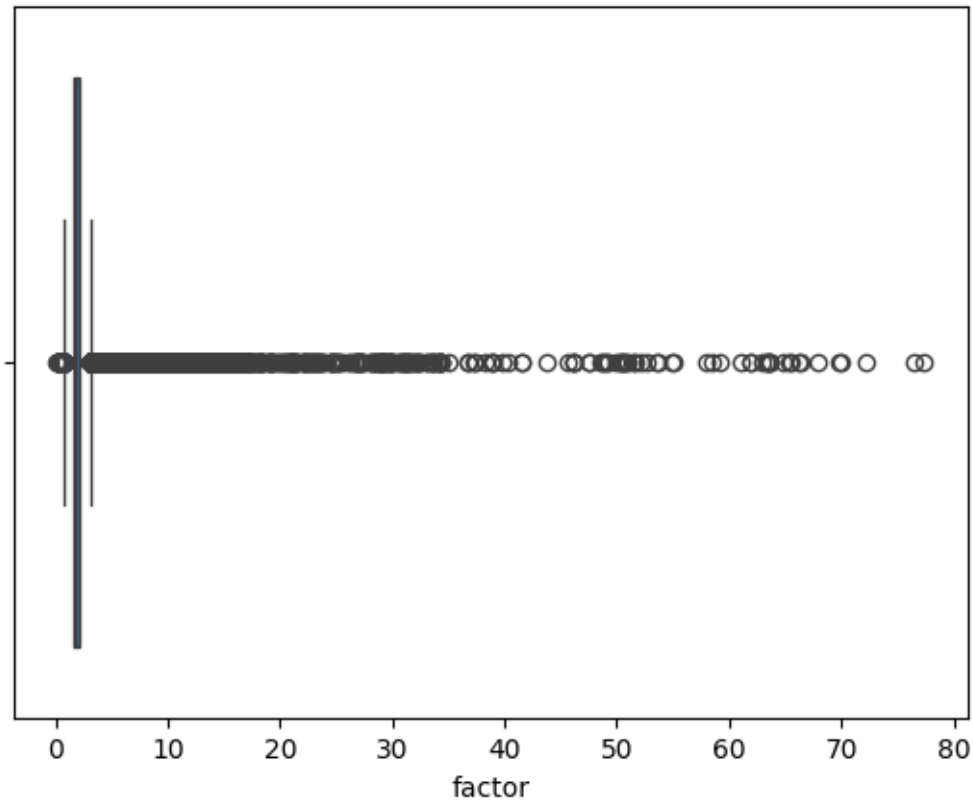

0
250
500
750
1000
1250
1500
1750
2000
cutoff_factor

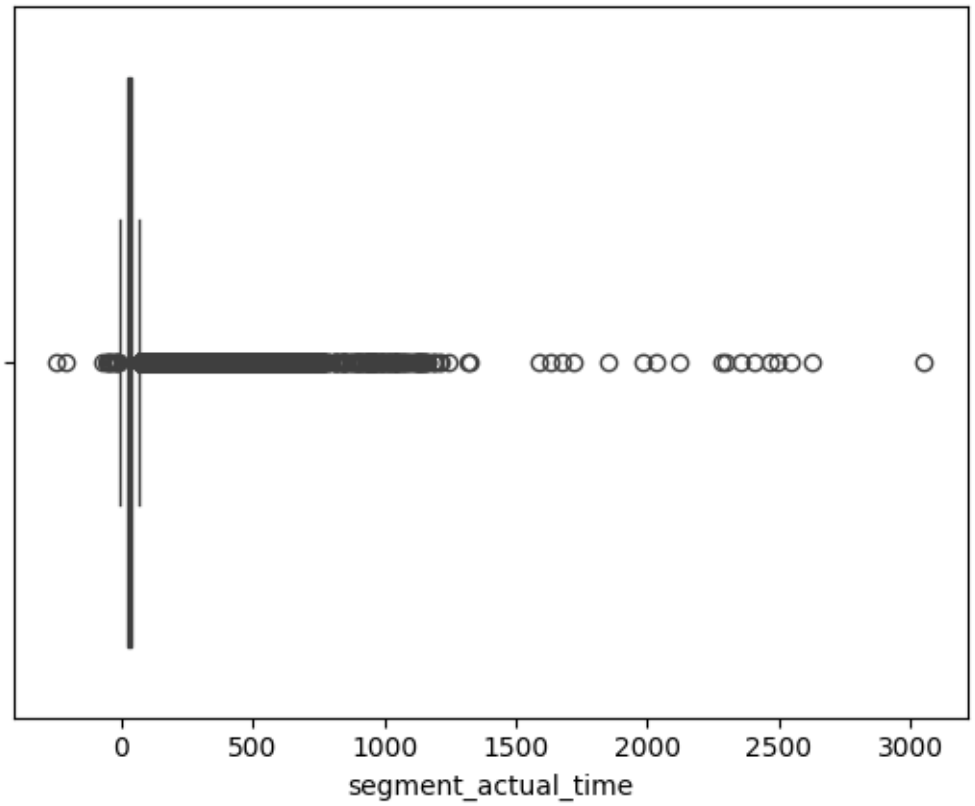

0
250
500
750
1000
1250
1500
1750
2000
actual_distance_to_destination

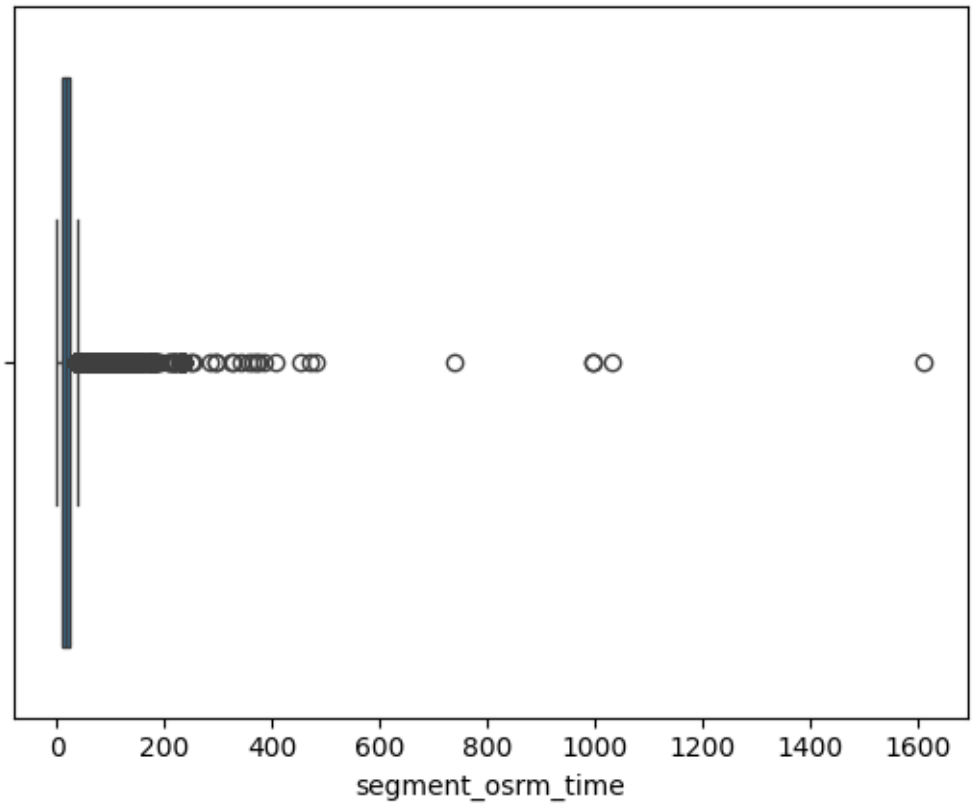

0
1000
2000
3000
4000
actual_time

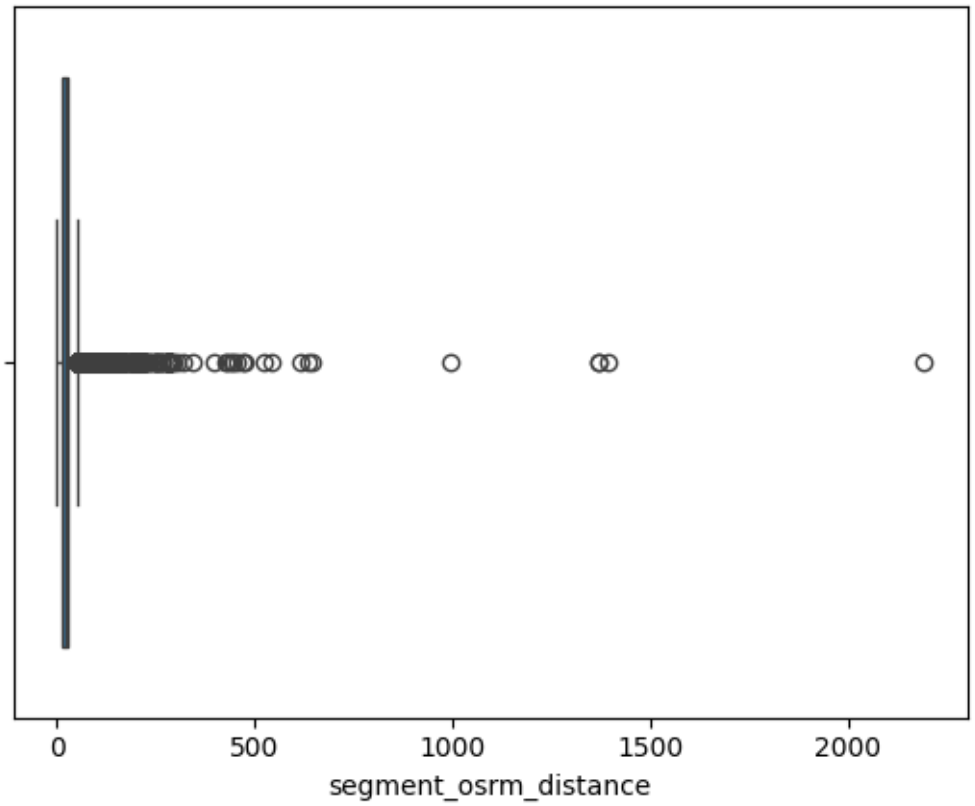

0
250
500
750
1000
1250
1500
1750
osrm_time

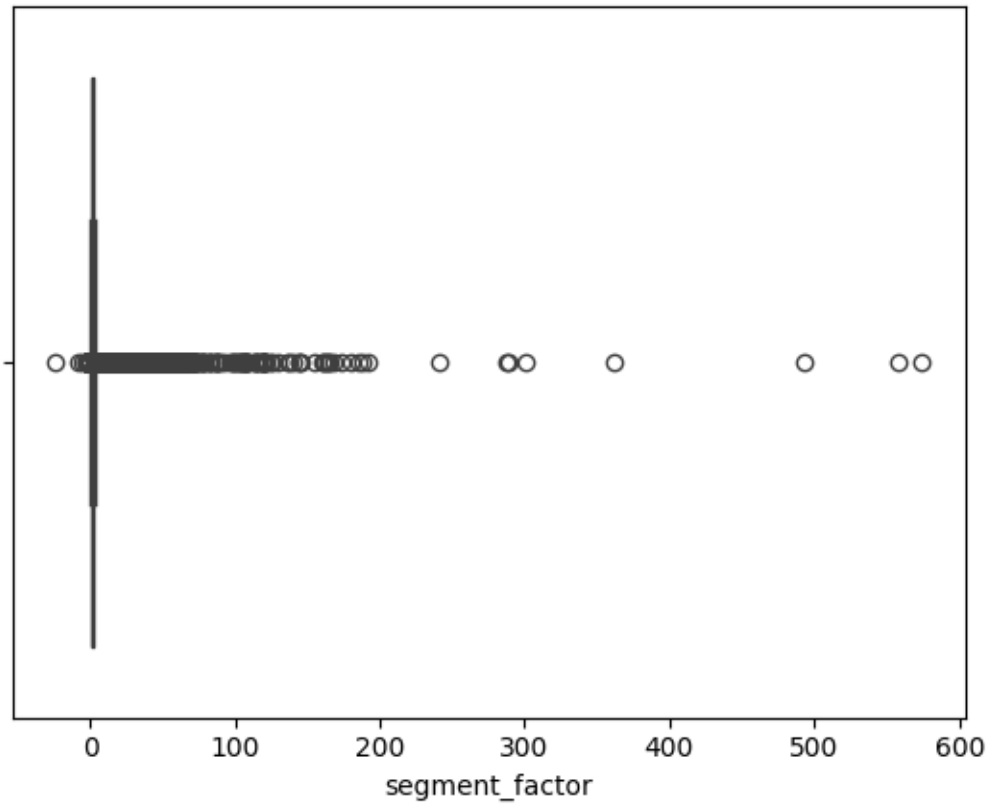

0
500
1000
1500
2000
osrm_distance


0
10
20
30
40
50
60
70
80
factor


0
500
1000
1500
2000
2500
3000
segment_actual_time


0
200
400
600
800
1000
1200
1400
1600
segment_osrm_time


0
500
1000
1500
2000
segment_osrm_distance



Try merging the rows using the hint mentioned above. Initial Grouping and Aggregation

[17]:
```
grouped = df_dropped.groupby(['trip_uuid', 'source_center',␣
↪'destination_center'])
grouped
df_dropped
```

[17]:
```
            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
3       training  2018-09-20 02:35:36.476840
4       training  2018-09-20 02:35:36.476840
…          …                          …
144862  training  2018-09-20 16:24:28.436231
144863  training  2018-09-20 16:24:28.436231
144864  training  2018-09-20 16:24:28.436231
144865  training  2018-09-20 16:24:28.436231
144866  training  2018-09-20 16:24:28.436231

                                    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
3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
…                                                     …       …
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    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)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
…                           …           …                       …
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)

       destination_center              destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
…                     …                           …
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                    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
3       2018-09-20 03:21:32.418600  …         2018-09-20 03:39:57
4       2018-09-20 03:21:32.418600  …         2018-09-20 03:33:55
…                              …  …                         …
144862  2018-09-20 16:24:28.436231  …         2018-09-20 21:57:20
144863  2018-09-20 16:24:28.436231  …         2018-09-20 21:31:18
```

```
144864  2018-09-20 16:24:28.436231  …         2018-09-20 21:11:18
144865  2018-09-20 16:24:28.436231  …         2018-09-20 20:53:19
144866  2018-09-20 16:24:28.436231  …  2018-09-20 16:24:28.436231

        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
3                            36.118028         62.0       40.0       45.5620
4                            39.386040         68.0       44.0       54.2181
…                                  …            …          …          …
144862                       45.258278         94.0       60.0       67.9280
144863                       54.092531        120.0       76.0       85.6829
144864                       66.163591        140.0       88.0       97.0933
144865                       73.680667        158.0       98.0      111.2709
144866                       70.039010        426.0       95.0       88.7319

          factor  segment_actual_time  segment_osrm_time  \
0       1.272727                 14.0               11.0
1       1.200000                 10.0                9.0
2       1.428571                 16.0                7.0
3       1.550000                 21.0               12.0
4       1.545455                  6.0                5.0
…          …                      …                  …
144862  1.566667                 12.0               12.0
144863  1.578947                 26.0               21.0
144864  1.590909                 20.0               34.0
144865  1.612245                 17.0               27.0
144866  4.484211                268.0                9.0

        segment_osrm_distance  segment_factor
0                     11.9653        1.272727
1                      9.7590        1.111111
2                     10.8152        2.285714
3                     13.0224        1.750000
4                      3.9153        1.200000
…                         …              …
144862                 8.1858        1.000000
144863                17.3725        1.238095
144864                20.7053        0.588235
144865                18.8885        0.629630
144866                 8.8088       29.777778

[144316 rows x 24 columns]
```

grouped = df_dropped.groupby([‘trip_uuid’, ‘source_center’, ‘destination_center’])

[18]:
```
grouped = df_dropped.groupby(['trip_uuid', 'source_center',␣
↪'destination_center']).agg({
    'od_start_time': 'first',    # Assuming od_start_time should take the␣
↪earliest time
    'od_end_time': 'last',     # Assuming od_end_time should take the latest␣
↪time
    'actual_distance_to_destination': 'sum',  'actual_time' :'sum'
})
grouped
```

[18]:
```
                                                od_start_time  \
trip_uuid                source_center destination_center
trip-153671041653548748 IND209304AAA  IND000000ACB        2018-09-12
16:39:46.858469
                        IND462022AAA  IND209304AAA        2018-09-12
00:00:16.535741
trip-153671042288605164 IND561203AAB  IND562101AAA        2018-09-12
02:03:09.655591
                        IND572101AAA  IND561203AAB        2018-09-12
00:00:22.886430
trip-153671043369099517 IND000000ACB  IND160002AAC        2018-09-14
03:40:17.106733
…
…
trip-153861115439069069 IND628204AAA  IND627657AAA        2018-10-04
02:29:04.272194
                        IND628613AAA  IND627005AAA        2018-10-04
04:16:39.894872
                        IND628801AAA  IND628204AAA        2018-10-04
01:44:53.808000
trip-153861118270144424 IND583119AAA  IND583101AAA        2018-10-04
03:58:40.726547
                        IND583201AAA  IND583119AAA        2018-10-04
02:51:44.712656

                                                od_end_time  \
trip_uuid                source_center destination_center
trip-153671041653548748 IND209304AAA  IND000000ACB        2018-09-13
13:40:23.123744
                        IND462022AAA  IND209304AAA        2018-09-12
16:39:46.858469
trip-153671042288605164 IND561203AAB  IND562101AAA        2018-09-12
03:01:59.598855
                        IND572101AAA  IND561203AAB        2018-09-12
02:03:09.655591
trip-153671043369099517 IND000000ACB  IND160002AAC        2018-09-14
17:34:55.442454
```

```
…
…
trip-153861115439069069 IND628204AAA  IND627657AAA        2018-10-04
03:31:11.183797
                        IND628613AAA  IND627005AAA        2018-10-04
05:47:45.162682
                        IND628801AAA  IND628204AAA        2018-10-04
02:29:04.272194
trip-153861118270144424 IND583119AAA  IND583101AAA        2018-10-04
08:46:09.166940
                        IND583201AAA  IND583119AAA        2018-10-04
03:58:40.726547

actual_distance_to_destination  \
trip_uuid               source_center destination_center
trip-153671041653548748 IND209304AAA  IND000000ACB
3778.765471
                        IND462022AAA  IND209304AAA
5082.046634
trip-153671042288605164 IND561203AAB  IND562101AAA
53.310332
                        IND572101AAA  IND561203AAB
186.897974
trip-153671043369099517 IND000000ACB  IND160002AAC
1725.590250
…
…
trip-153861115439069069 IND628204AAA  IND627657AAA
88.326510
                        IND628613AAA  IND627005AAA
90.049767
                        IND628801AAA  IND628204AAA
21.672374
trip-153861118270144424 IND583119AAA  IND583101AAA
62.547507
                        IND583201AAA  IND583119AAA
47.691610

                                                    actual_time
trip_uuid               source_center destination_center
trip-153671041653548748 IND209304AAA  IND000000ACB        6484.0
                        IND462022AAA  IND209304AAA        9198.0
trip-153671042288605164 IND561203AAB  IND562101AAA          96.0
                        IND572101AAA  IND561203AAB         303.0
trip-153671043369099517 IND000000ACB  IND160002AAC        2601.0
…                                                            …
trip-153861115439069069 IND628204AAA  IND627657AAA         119.0
```

```
                          IND628613AAA  IND627005AAA              173.0
                          IND628801AAA  IND628204AAA               51.0
trip-153861118270144424 IND583119AAA  IND583101AAA              278.0
                          IND583201AAA  IND583119AAA               72.0

[26222 rows x 4 columns]
```

Aggregation

[19]:
```
# Convert necessary fields to datetime if they are not already
df_dropped['trip_creation_time'] = pd.
 ↪to_datetime(df_dropped['trip_creation_time'])
df_dropped['od_start_time'] = pd.to_datetime(df_dropped['od_start_time'])
df_dropped['od_end_time'] = pd.to_datetime(df_dropped['od_end_time'])

# First level of aggregation: Trip_uuid, Source ID, Destination ID
aggregated_df = df_dropped.groupby(['trip_uuid', 'source_center',␣
 ↪'destination_center']).agg({
    'actual_time': 'sum',  # Summing up times and distances
    'osrm_time': 'sum',
    'actual_distance_to_destination': 'sum',
    'osrm_distance': 'sum',
    'trip_creation_time': 'first',  # Assuming the earliest trip creation time␣
 ↪is what we need
    'od_start_time': 'first',
    'od_end_time': 'last',  # Last od_end_time as the final end time
    'source_name': 'first',  # First source name (start of the journey)
    'destination_name': 'last',  # Last destination name (end of the journey)
    'route_type': 'first'  # Route type might be consistent across segments,␣
 ↪taking first
}).reset_index()

# Second level of aggregation: Just Trip_uuid
final_aggregated_df = aggregated_df.groupby('trip_uuid').agg({
    'actual_time': 'sum',
    'osrm_time': 'sum',
    'actual_distance_to_destination': 'sum',
    'osrm_distance': 'sum',
    'trip_creation_time': 'first',
    'od_start_time': 'first',
    'od_end_time': 'last',
    'source_name': 'first',
    'destination_name': 'last',
    'route_type': 'first'
}).reset_index()

# Checking the aggregated data
```

```
final_aggregated_df.head()
```

```
<ipython-input-19-c423ab17b26a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped['trip_creation_time'] =
pd.to_datetime(df_dropped['trip_creation_time'])
<ipython-input-19-c423ab17b26a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped['od_start_time'] = pd.to_datetime(df_dropped['od_start_time'])
<ipython-input-19-c423ab17b26a>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped['od_end_time'] = pd.to_datetime(df_dropped['od_end_time'])
```

[19]:

	trip_uuid	actual_time	osrm_time	\
0	trip-153671041653548748	15682.0	7787.0	
1	trip-153671042288605164	399.0	210.0	
2	trip-153671043369099517	112225.0	65768.0	
3	trip-153671046011330457	82.0	24.0	
4	trip-153671052974046625	556.0	207.0	

	actual_distance_to_destination	osrm_distance	trip_creation_time	\
0	8860.812105	10577.7647	2018-09-12 00:00:16.535741	
1	240.208306	269.4308	2018-09-12 00:00:22.886430	
2	68163.502238	89447.2488	2018-09-12 00:00:33.691250	
3	28.529648	31.6475	2018-09-12 00:01:00.113710	
4	239.007304	266.2914	2018-09-12 00:02:09.740725	

	od_start_time	od_end_time	\
0	2018-09-12 16:39:46.858469	2018-09-12 16:39:46.858469	
1	2018-09-12 02:03:09.655591	2018-09-12 02:03:09.655591	
2	2018-09-14 03:40:17.106733	2018-09-14 03:40:17.106733	
3	2018-09-12 00:01:00.113710	2018-09-12 01:41:29.809822	
4	2018-09-12 00:02:09.740725	2018-09-12 03:54:43.114421	

	source_name	destination_name	\

```
0  Kanpur_Central_H_6 (Uttar Pradesh)  Kanpur_Central_H_6 (Uttar Pradesh)
1   Doddablpur_ChikaDPP_D (Karnataka)   Doddablpur_ChikaDPP_D (Karnataka)
2       Gurgaon_Bilaspur_HB (Haryana)       Gurgaon_Bilaspur_HB (Haryana)
3            Mumbai Hub (Maharashtra)      Mumbai_MiraRd_IP (Maharashtra)
4              Bellary_Dc (Karnataka)       Sandur_WrdN1DPP_D (Karnataka)

  route_type
0        FTL
1    Carting
2        FTL
3    Carting
4        FTL
```

[20]:
```
final_aggregated_df.columns
```

[20]:
```
Index(['trip_uuid', 'actual_time', 'osrm_time',
       'actual_distance_to_destination', 'osrm_distance', 'trip_creation_time',
       'od_start_time', 'od_end_time', 'source_name', 'destination_name',
       'route_type'],
      dtype='object')
```

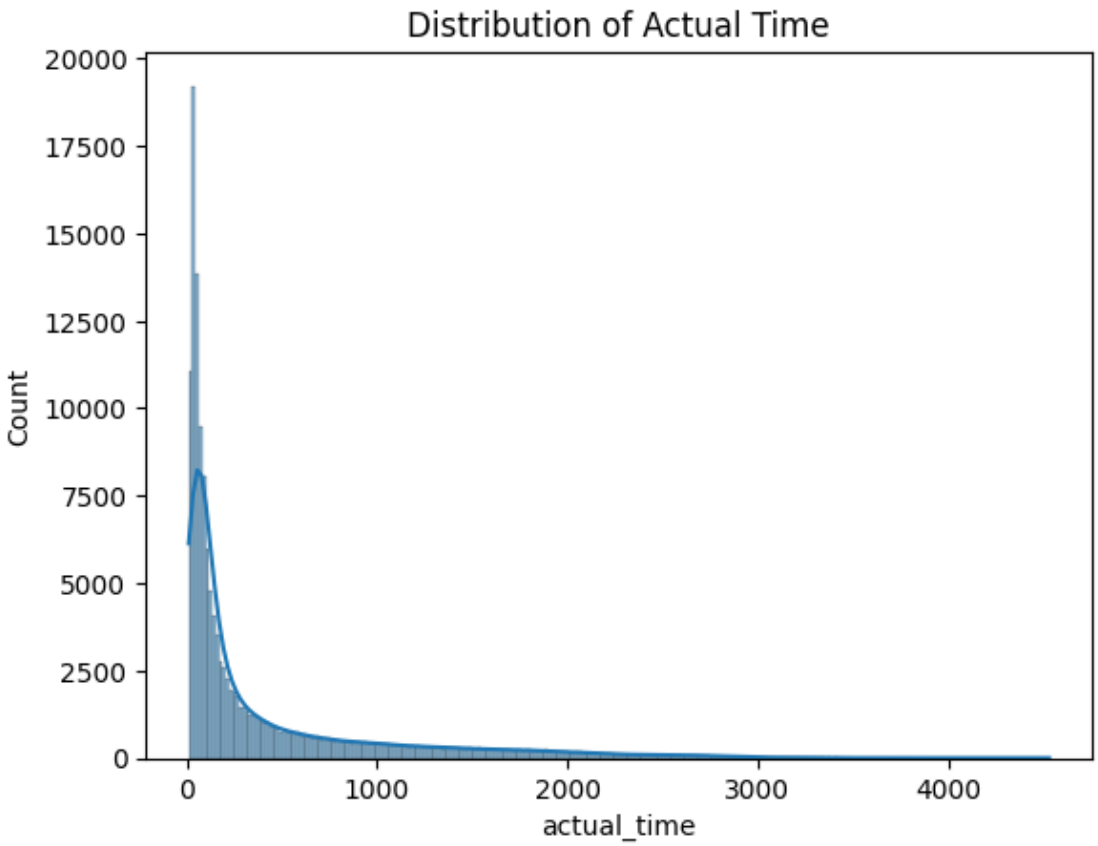
2.0.1 Range of numerical data

[21]:
```
df['actual_time'] = pd.to_numeric(df['actual_time'], errors='coerce')
min_time = df['actual_time'].min()
max_time = df['actual_time'].max()

print(f"The range of actual_time is from {min_time} to {max_time}")
```

```
The range of actual_time is from 9.0 to 4532.0
```

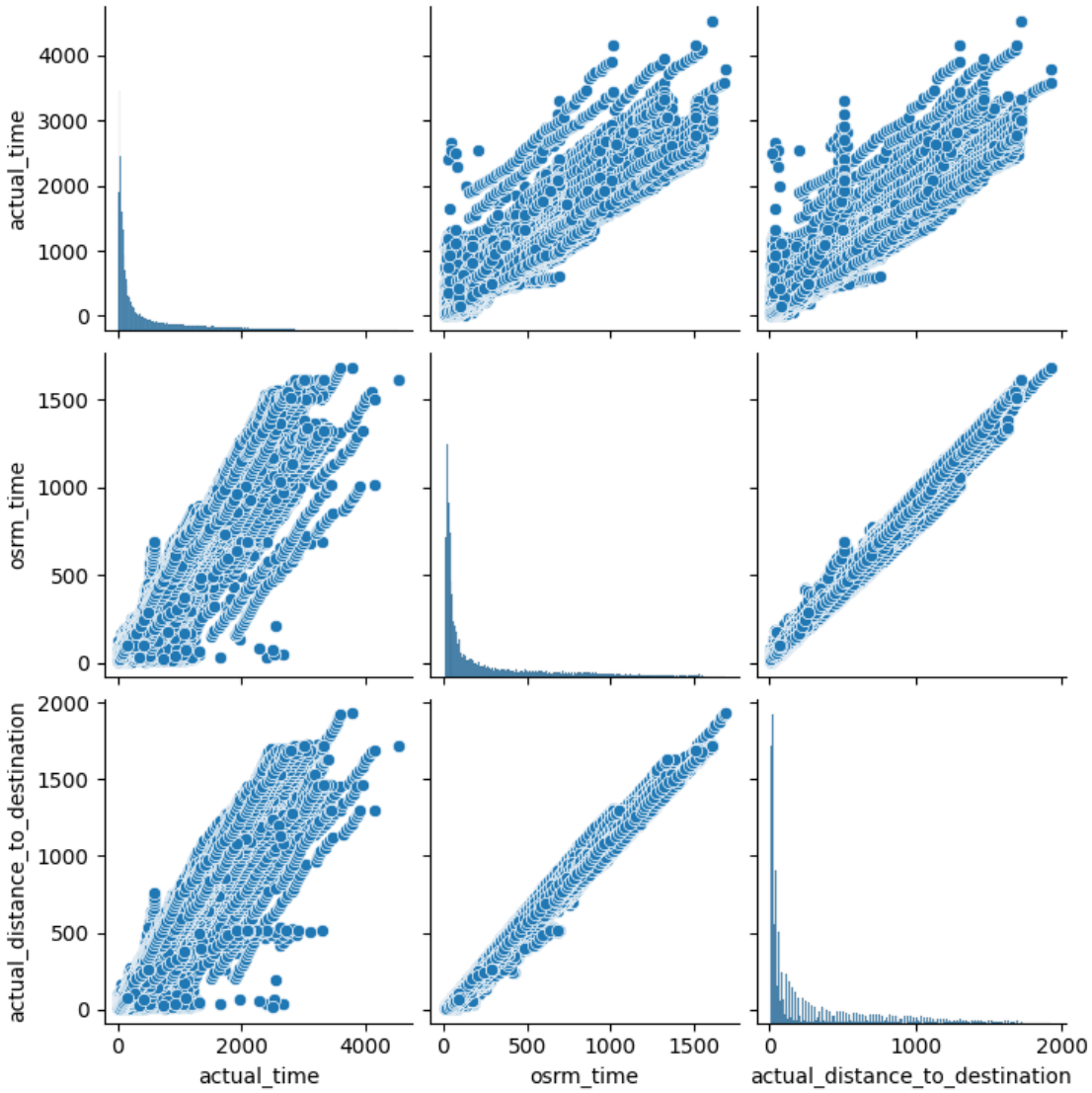
2.0.2 Univariate Analysis

[22]:
```
sns.histplot(df_dropped['actual_time'], kde=True)
plt.title('Distribution of Actual Time')
plt.show()
```



2.0.3 Bivariate Analysis

[23]:
```
sns.pairplot(df_dropped[['actual_time', 'osrm_time',␣
↪'actual_distance_to_destination']])
plt.show()
```



3 2. Feature Creation (10 Points)

[28]:
```
dd = df_dropped
#dd = final_aggregated_df
dd
```

[28]:
```
            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
3       training 2018-09-20 02:35:36.476840
4       training 2018-09-20 02:35:36.476840
```

```
…          …                          …
144862  training 2018-09-20 16:24:28.436231
144863  training 2018-09-20 16:24:28.436231
144864  training 2018-09-20 16:24:28.436231
144865  training 2018-09-20 16:24:28.436231
144866  training 2018-09-20 16:24:28.436231

                                    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
3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
…                                                     …        …
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    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)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
…                           …           …                          …
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)

       destination_center               destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
…                     …                           …
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                  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
3      2018-09-20 03:21:32.418600  …         2018-09-20 03:39:57
4      2018-09-20 03:21:32.418600  …         2018-09-20 03:33:55
…                              …  …                         …
144862 2018-09-20 16:24:28.436231  …         2018-09-20 21:57:20
144863 2018-09-20 16:24:28.436231  …         2018-09-20 21:31:18
144864 2018-09-20 16:24:28.436231  …         2018-09-20 21:11:18
144865 2018-09-20 16:24:28.436231  …         2018-09-20 20:53:19
144866 2018-09-20 16:24:28.436231  …  2018-09-20 16:24:28.436231

        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
3                            36.118028         62.0       40.0       45.5620
4                            39.386040         68.0       44.0       54.2181
…                                  …            …          …          …
144862                       45.258278         94.0       60.0       67.9280
144863                       54.092531        120.0       76.0       85.6829
144864                       66.163591        140.0       88.0       97.0933
144865                       73.680667        158.0       98.0      111.2709
144866                       70.039010        426.0       95.0       88.7319

          factor  segment_actual_time  segment_osrm_time  \
0       1.272727                 14.0               11.0
1       1.200000                 10.0                9.0
2       1.428571                 16.0                7.0
3       1.550000                 21.0               12.0
4       1.545455                  6.0                5.0
…          …                      …                  …
144862  1.566667                 12.0               12.0
144863  1.578947                 26.0               21.0
144864  1.590909                 20.0               34.0
144865  1.612245                 17.0               27.0
144866  4.484211                268.0                9.0

        segment_osrm_distance  segment_factor
0                     11.9653        1.272727
1                      9.7590        1.111111
2                     10.8152        2.285714
3                     13.0224        1.750000
4                      3.9153        1.200000
…                         …             …
144862                 8.1858        1.000000
144863                17.3725        1.238095
```

```
144864                20.7053         0.588235
144865                18.8885         0.629630
144866                 8.8088        29.777778

[144316 rows x 24 columns]
```

[29]:
```
def extract_features(name):
    parts = name.split('_')
    city = '_'.join(parts[:-1])
    place_code = parts[-1].split(' ')[0]
    state = parts[-1].split('(')[-1].strip(')')
    return city, place_code, state
```

[30]:
```
df_clean = dd.copy(deep = True)
df_clean_aggregate = final_aggregated_df
```

[31]:
```
# Apply feature extraction to Destination Name
df_clean_aggregate[['destination_city', 'destination_place_code',␣
 ↪'destination_state']] = df_clean_aggregate['destination_name'].apply(
    lambda x: pd.Series(extract_features(x))
)

# Apply feature extraction to Source Name
df_clean_aggregate[['source_city', 'source_place_code', 'source_state']] =␣
 ↪df_clean_aggregate['source_name'].apply(
    lambda x: pd.Series(extract_features(x))
)

df_clean_aggregate['trip_creation_time'] = pd.
 ↪to_datetime(df_clean_aggregate['trip_creation_time'], format='%Y-%m-%d %H:%M:
 ↪%S.%f')

# Extract features
df_clean_aggregate['Year'] = df_clean_aggregate['trip_creation_time'].dt.year
df_clean_aggregate['Month'] = df_clean_aggregate['trip_creation_time'].dt.month
df_clean_aggregate['Day'] = df_clean_aggregate['trip_creation_time'].dt.day

# Print the dataframe with extracted features
df_clean_aggregate.head()
```

[31]:
```
                 trip_uuid  actual_time  osrm_time  \
0  trip-153671041653548748      15682.0     7787.0
1  trip-153671042288605164        399.0      210.0
2  trip-153671043369099517     112225.0    65768.0
3  trip-153671046011330457         82.0       24.0
4  trip-153671052974046625        556.0      207.0
```

```
   actual_distance_to_destination  osrm_distance         trip_creation_time  \
0                     8860.812105     10577.7647 2018-09-12 00:00:16.535741
1                      240.208306       269.4308 2018-09-12 00:00:22.886430
2                    68163.502238     89447.2488 2018-09-12 00:00:33.691250
3                       28.529648        31.6475 2018-09-12 00:01:00.113710
4                      239.007304       266.2914 2018-09-12 00:02:09.740725

               od_start_time                od_end_time  \
0 2018-09-12 16:39:46.858469 2018-09-12 16:39:46.858469
1 2018-09-12 02:03:09.655591 2018-09-12 02:03:09.655591
2 2018-09-14 03:40:17.106733 2018-09-14 03:40:17.106733
3 2018-09-12 00:01:00.113710 2018-09-12 01:41:29.809822
4 2018-09-12 00:02:09.740725 2018-09-12 03:54:43.114421

                         source_name                    destination_name  \
0  Kanpur_Central_H_6 (Uttar Pradesh)  Kanpur_Central_H_6 (Uttar Pradesh)
1   Doddablpur_ChikaDPP_D (Karnataka)   Doddablpur_ChikaDPP_D (Karnataka)
2       Gurgaon_Bilaspur_HB (Haryana)       Gurgaon_Bilaspur_HB (Haryana)
3            Mumbai Hub (Maharashtra)      Mumbai_MiraRd_IP (Maharashtra)
4              Bellary_Dc (Karnataka)       Sandur_WrdN1DPP_D (Karnataka)

  route_type     destination_city destination_place_code destination_state  \
0        FTL     Kanpur_Central_H                      6     Uttar Pradesh
1    Carting  Doddablpur_ChikaDPP                      D         Karnataka
2        FTL     Gurgaon_Bilaspur                     HB           Haryana
3    Carting        Mumbai_MiraRd                     IP       Maharashtra
4        FTL      Sandur_WrdN1DPP                      D         Karnataka

           source_city source_place_code   source_state  Year  Month  Day
0     Kanpur_Central_H                6  Uttar Pradesh  2018      9   12
1  Doddablpur_ChikaDPP                D      Karnataka  2018      9   12
2     Gurgaon_Bilaspur               HB        Haryana  2018      9   12
3                               Mumbai    Maharashtra  2018      9   12
4              Bellary               Dc      Karnataka  2018      9   12
```

[33]:
```python
# Apply feature extraction to Destination Name
df_clean[['destination_city', 'destination_place_code', 'destination_state']] =␣
 ↪df_clean['destination_name'].apply(
    lambda x: pd.Series(extract_features(x))
)

# Apply feature extraction to Source Name
df_clean[['source_city', 'source_place_code', 'source_state']] =␣
 ↪df_clean['source_name'].apply(
    lambda x: pd.Series(extract_features(x))
)
```

```
df_clean.head()
```

[33]:
```
       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
3  training 2018-09-20 02:35:36.476840
4  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
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4  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)
3  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4  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)
3       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)

              od_start_time  … segment_actual_time  segment_osrm_time  \
0 2018-09-20 03:21:32.418600  …                14.0               11.0
1 2018-09-20 03:21:32.418600  …                10.0                9.0
2 2018-09-20 03:21:32.418600  …                16.0                7.0
3 2018-09-20 03:21:32.418600  …                21.0               12.0
4 2018-09-20 03:21:32.418600  …                 6.0                5.0

   segment_osrm_distance  segment_factor   destination_city  \
0                11.9653        1.272727  Khambhat_MotvdDPP
1                 9.7590        1.111111  Khambhat_MotvdDPP
2                10.8152        2.285714  Khambhat_MotvdDPP
3                13.0224        1.750000  Khambhat_MotvdDPP
4                 3.9153        1.200000  Khambhat_MotvdDPP

   destination_place_code  destination_state    source_city  \
0                       D            Gujarat  Anand_VUNagar
1                       D            Gujarat  Anand_VUNagar
```

```
2                       D           Gujarat  Anand_VUNagar
3                       D           Gujarat  Anand_VUNagar
4                       D           Gujarat  Anand_VUNagar

   source_place_code  source_state
0                 DC       Gujarat
1                 DC       Gujarat
2                 DC       Gujarat
3                 DC       Gujarat
4                 DC       Gujarat

[5 rows x 30 columns]
```

[34]:
```
df_clean['trip_creation_time'] = pd.to_datetime(df_clean['trip_creation_time'],␣
 ↪format='%Y-%m-%d %H:%M:%S.%f')

# Extract features
df_clean['Year'] = df_clean['trip_creation_time'].dt.year
df_clean['Month'] = df_clean['trip_creation_time'].dt.month
df_clean['Day'] = df_clean['trip_creation_time'].dt.day

# Print the dataframe with extracted features
df_clean[['trip_creation_time', 'Year', 'Month', 'Day']]
```

[34]:
```
                trip_creation_time  Year  Month  Day
0      2018-09-20 02:35:36.476840  2018      9   20
1      2018-09-20 02:35:36.476840  2018      9   20
2      2018-09-20 02:35:36.476840  2018      9   20
3      2018-09-20 02:35:36.476840  2018      9   20
4      2018-09-20 02:35:36.476840  2018      9   20
…                             …     …      …   …
144862 2018-09-20 16:24:28.436231  2018      9   20
144863 2018-09-20 16:24:28.436231  2018      9   20
144864 2018-09-20 16:24:28.436231  2018      9   20
144865 2018-09-20 16:24:28.436231  2018      9   20
144866 2018-09-20 16:24:28.436231  2018      9   20

[144316 rows x 4 columns]
```

[35]:
```
df_clean.head()
```

[35]:
```
       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
3  training 2018-09-20 02:35:36.476840
4  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
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4  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)
3  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4  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)
3       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)

              od_start_time  … segment_factor   destination_city  \
0 2018-09-20 03:21:32.418600  …       1.272727  Khambhat_MotvdDPP
1 2018-09-20 03:21:32.418600  …       1.111111  Khambhat_MotvdDPP
2 2018-09-20 03:21:32.418600  …       2.285714  Khambhat_MotvdDPP
3 2018-09-20 03:21:32.418600  …       1.750000  Khambhat_MotvdDPP
4 2018-09-20 03:21:32.418600  …       1.200000  Khambhat_MotvdDPP

   destination_place_code  destination_state    source_city  \
0                       D            Gujarat  Anand_VUNagar
1                       D            Gujarat  Anand_VUNagar
2                       D            Gujarat  Anand_VUNagar
3                       D            Gujarat  Anand_VUNagar
4                       D            Gujarat  Anand_VUNagar

   source_place_code  source_state  Year  Month  Day
0                 DC       Gujarat  2018      9   20
1                 DC       Gujarat  2018      9   20
2                 DC       Gujarat  2018      9   20
3                 DC       Gujarat  2018      9   20
4                 DC       Gujarat  2018      9   20

[5 rows x 33 columns]
```

4 3. Merging of rows and aggregation of fields (10 Points)

[36]:
```
# Convert necessary fields to datetime if they are not already
df_dropped['trip_creation_time'] = pd.
 ↪to_datetime(df_dropped['trip_creation_time'])
df_dropped['od_start_time'] = pd.to_datetime(df_dropped['od_start_time'])
df_dropped['od_end_time'] = pd.to_datetime(df_dropped['od_end_time'])

# First level of aggregation: Trip_uuid, Source ID, Destination ID
aggregated_df = df_dropped.groupby(['trip_uuid', 'source_center',␣
 ↪'destination_center']).agg({
    'actual_time': 'sum',  # Summing up times and distances
    'osrm_time': 'sum',
    'actual_distance_to_destination': 'sum',
    'osrm_distance': 'sum',
    'trip_creation_time': 'first',  # Assuming the earliest trip creation time␣
 ↪is what we need
    'od_start_time': 'first',
    'od_end_time': 'last',  # Last od_end_time as the final end time
    'source_name': 'first',  # First source name (start of the journey)
    'destination_name': 'last',  # Last destination name (end of the journey)
    'route_type': 'first'  # Route type might be consistent across segments,␣
 ↪taking first
}).reset_index()

# Second level of aggregation: Just Trip_uuid
final_aggregated_df = aggregated_df.groupby('trip_uuid').agg({
    'actual_time': 'sum',
    'osrm_time': 'sum',
    'actual_distance_to_destination': 'sum',
    'osrm_distance': 'sum',
    'trip_creation_time': 'first',
    'od_start_time': 'first',
    'od_end_time': 'last',
    'source_name': 'first',
    'destination_name': 'last',
    'route_type': 'first'
}).reset_index()

# Checking the aggregated data
final_aggregated_df.head()
```

```
<ipython-input-36-c423ab17b26a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
```

```
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped['trip_creation_time'] =
pd.to_datetime(df_dropped['trip_creation_time'])
<ipython-input-36-c423ab17b26a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped['od_start_time'] = pd.to_datetime(df_dropped['od_start_time'])
<ipython-input-36-c423ab17b26a>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_dropped['od_end_time'] = pd.to_datetime(df_dropped['od_end_time'])
```

[36]:

	trip_uuid	actual_time	osrm_time	\
0	trip-153671041653548748	15682.0	7787.0	
1	trip-153671042288605164	399.0	210.0	
2	trip-153671043369099517	112225.0	65768.0	
3	trip-153671046011330457	82.0	24.0	
4	trip-153671052974046625	556.0	207.0	

	actual_distance_to_destination	osrm_distance	trip_creation_time	\
0	8860.812105	10577.7647	2018-09-12 00:00:16.535741	
1	240.208306	269.4308	2018-09-12 00:00:22.886430	
2	68163.502238	89447.2488	2018-09-12 00:00:33.691250	
3	28.529648	31.6475	2018-09-12 00:01:00.113710	
4	239.007304	266.2914	2018-09-12 00:02:09.740725	

	od_start_time	od_end_time	\
0	2018-09-12 16:39:46.858469	2018-09-12 16:39:46.858469	
1	2018-09-12 02:03:09.655591	2018-09-12 02:03:09.655591	
2	2018-09-14 03:40:17.106733	2018-09-14 03:40:17.106733	
3	2018-09-12 00:01:00.113710	2018-09-12 01:41:29.809822	
4	2018-09-12 00:02:09.740725	2018-09-12 03:54:43.114421	

	source_name	destination_name	\
0	Kanpur_Central_H_6 (Uttar Pradesh)	Kanpur_Central_H_6 (Uttar Pradesh)	
1	Doddablpur_ChikaDPP_D (Karnataka)	Doddablpur_ChikaDPP_D (Karnataka)	
2	Gurgaon_Bilaspur_HB (Haryana)	Gurgaon_Bilaspur_HB (Haryana)	
3	Mumbai Hub (Maharashtra)	Mumbai_MiraRd_IP (Maharashtra)	
4	Bellary_Dc (Karnataka)	Sandur_WrdN1DPP_D (Karnataka)	

	route_type

```
0        FTL
1    Carting
2        FTL
3    Carting
4        FTL
```

[36]:

5 QUESTION 3

In-depth analysis and feature engineering:

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

[37]:
```
dd = df_clean
dd['od_start_time'] = pd.to_datetime(dd['od_start_time'], format='%Y-%m-%d %H:
 ↪%M:%S.%f', errors='coerce')
dd['od_end_time'] = pd.to_datetime(dd['od_end_time'], format='%Y-%m-%d %H:%M:%S.
 ↪%f', errors='coerce')

# Calculate the time difference in minutes
dd['time_difference'] = (dd['od_end_time'] - dd['od_start_time']).dt.seconds /␣
 ↪60

# Print the DataFrame to verify the changes
dd
```

[37]:
```
            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
3       training 2018-09-20 02:35:36.476840
4       training 2018-09-20 02:35:36.476840
…          …                          …
144862  training 2018-09-20 16:24:28.436231
144863  training 2018-09-20 16:24:28.436231
144864  training 2018-09-20 16:24:28.436231
144865  training 2018-09-20 16:24:28.436231
144866  training 2018-09-20 16:24:28.436231

                                    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
3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
```

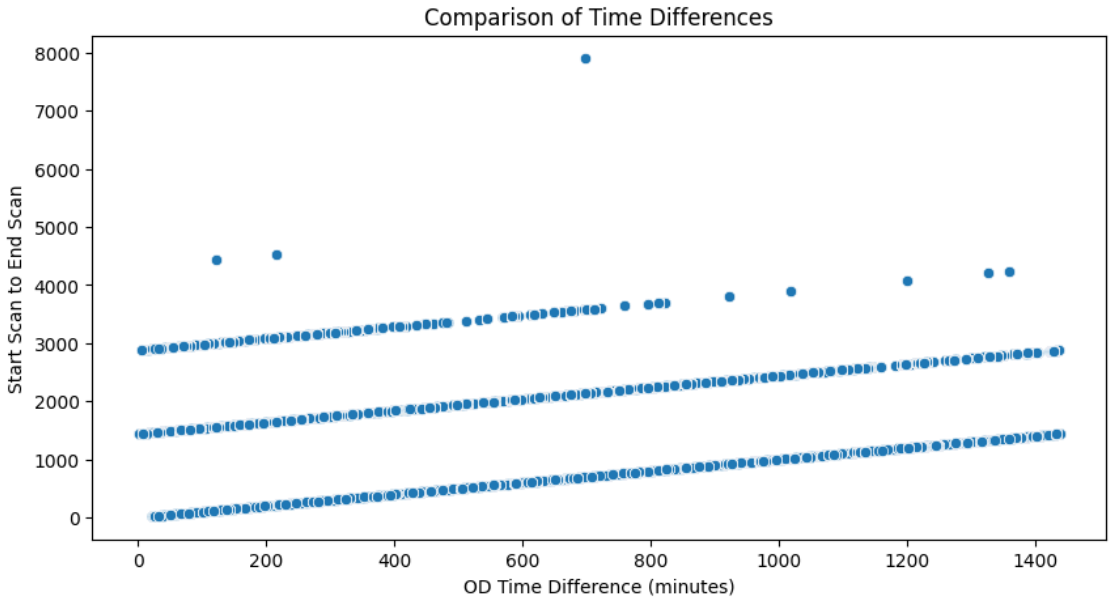
```
…                                                     …        …
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    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)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
…                           …             …                          …
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)

       destination_center              destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
…                       …                           …
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                   od_start_time  …   destination_city  \
0      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
1      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
2      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
3      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
4      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
…                             …  …                …
144862 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144863 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144864 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144865 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144866 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur

        destination_place_code  destination_state     source_city  \
```

```
0                            D            Gujarat   Anand_VUNagar
1                            D            Gujarat   Anand_VUNagar
2                            D            Gujarat   Anand_VUNagar
3                            D            Gujarat   Anand_VUNagar
4                            D            Gujarat   Anand_VUNagar
…                          …              …              …
144862                      HB            Haryana  Sonipat_Kundli
144863                      HB            Haryana  Sonipat_Kundli
144864                      HB            Haryana  Sonipat_Kundli
144865                      HB            Haryana  Sonipat_Kundli
144866                      HB            Haryana  Sonipat_Kundli

       source_place_code  source_state  Year  Month  Day  time_difference
0                     DC       Gujarat  2018      9   20        86.200000
1                     DC       Gujarat  2018      9   20        86.200000
2                     DC       Gujarat  2018      9   20        86.200000
3                     DC       Gujarat  2018      9   20        86.200000
4                     DC       Gujarat  2018      9   20        86.200000
…                   …            …    …   …  …              …
144862                 H       Haryana  2018      9   20       427.683333
144863                 H       Haryana  2018      9   20       427.683333
144864                 H       Haryana  2018      9   20       427.683333
144865                 H       Haryana  2018      9   20       427.683333
144866                 H       Haryana  2018      9   20       427.683333

[144316 rows x 34 columns]
```

6 4. Comparison & Visualization of time and distance fields

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

[38]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting
plt.figure(figsize=(10, 5))
sns.scatterplot(x=dd['time_difference'], y=dd['start_scan_to_end_scan'])
plt.title('Comparison of Time Differences')
plt.xlabel('OD Time Difference (minutes)')
plt.ylabel('Start Scan to End Scan')
plt.show()
```



7 Hypothesis Testing/Visual Analysis

Aggregated values after merging the rows on the basis of trip_uuid

[39]:
```
aggregated = dd.groupby('trip_uuid').agg({
    'actual_time': 'mean',
    'osrm_time': 'mean',
    'segment_actual_time': 'mean',
    'osrm_distance': 'mean',
    'segment_osrm_distance': 'mean',
    'osrm_time': 'mean',
    'segment_osrm_time': 'mean'
}).reset_index()
```

[40]:
```
dd.head()
```

[40]:

	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	
3	training	2018-09-20 02:35:36.476840	
4	training	2018-09-20 02:35:36.476840	

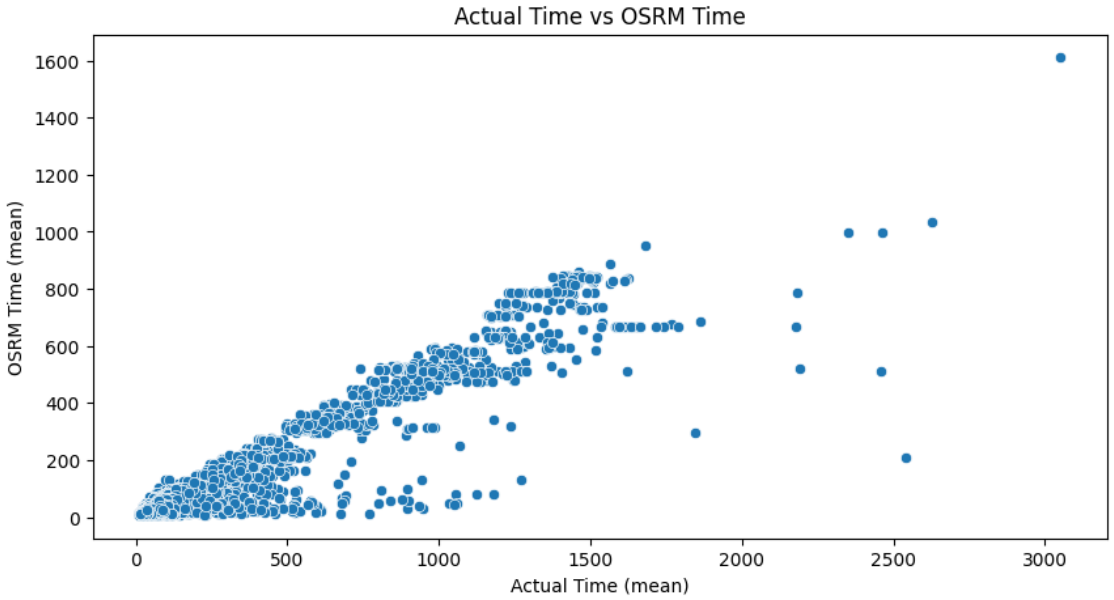
	route_schedule_uuid	route_type	\
0	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	
1	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	Carting	

```
2  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4  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)
3  trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4  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)
3       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4       IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)

              od_start_time  …   destination_city  destination_place_code  \
0 2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP                       D
1 2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP                       D
2 2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP                       D
3 2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP                       D
4 2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP                       D

   destination_state    source_city source_place_code  source_state  Year  \
0           Gujarat  Anand_VUNagar                DC       Gujarat  2018
1           Gujarat  Anand_VUNagar                DC       Gujarat  2018
2           Gujarat  Anand_VUNagar                DC       Gujarat  2018
3           Gujarat  Anand_VUNagar                DC       Gujarat  2018
4           Gujarat  Anand_VUNagar                DC       Gujarat  2018

   Month  Day  time_difference
0      9   20             86.2
1      9   20             86.2
2      9   20             86.2
3      9   20             86.2
4      9   20             86.2

[5 rows x 34 columns]
```

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)

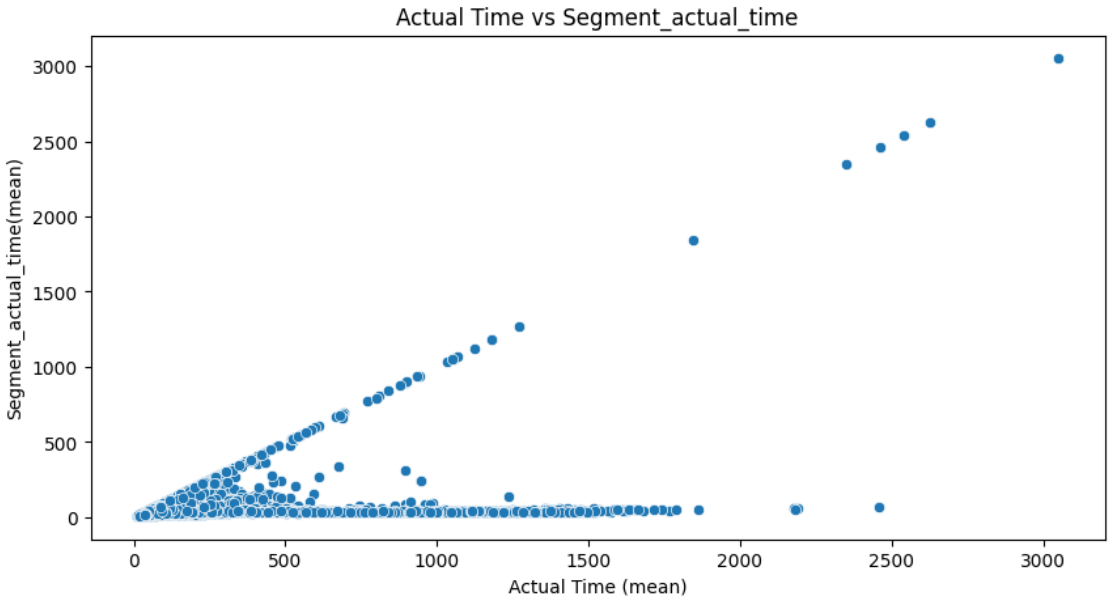
[41]:
```
plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['actual_time'], y=aggregated['osrm_time'])
```

```
plt.title('Actual Time vs OSRM Time')
plt.xlabel('Actual Time (mean)')
plt.ylabel('OSRM Time (mean)')
plt.show()
```



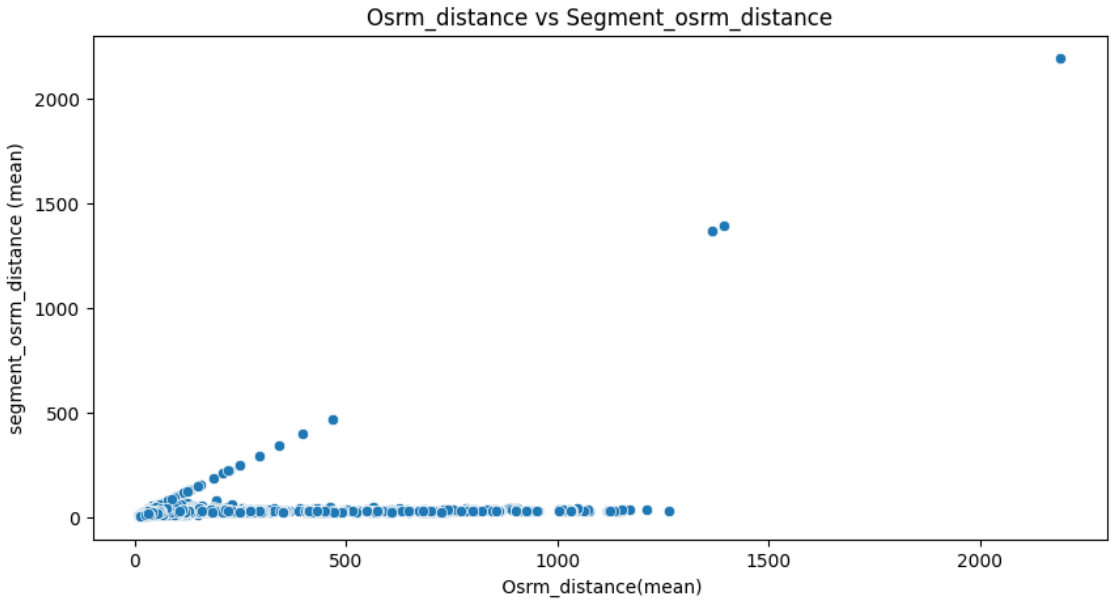
Do hypothesis testing/ visual analysis between actual_time aggregated value and segment actual time aggregated value (aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)

[42]:
```
plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['actual_time'],␣
  ↪y=aggregated['segment_actual_time'])
plt.title('Actual Time vs Segment_actual_time')
plt.xlabel('Actual Time (mean)')
plt.ylabel('Segment_actual_time(mean)')
plt.show()
```



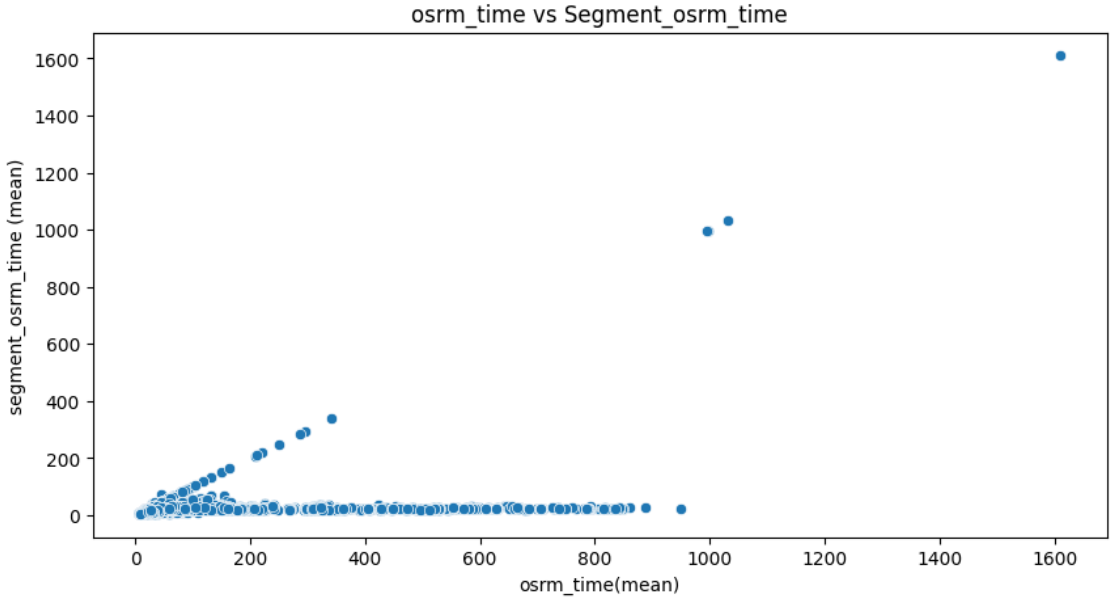
Do hypothesis testing/ visual analysis between osrm distance aggregated value and segment osrm distance aggregated value (aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)

[43]:
```
plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['osrm_distance'],␣
↪y=aggregated['segment_osrm_distance'])
plt.title('Osrm_distance vs Segment_osrm_distance')
plt.xlabel(' Osrm_distance(mean)')
plt.ylabel('segment_osrm_distance (mean)')
plt.show()
```



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)

[44]:
```
plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['osrm_time'], y=aggregated['segment_osrm_time'])
plt.title('osrm_time vs Segment_osrm_time')
plt.xlabel('osrm_time(mean)')
plt.ylabel('segment_osrm_time (mean)')
plt.show()
```



8 5. Missing values Treatment & Outlier treatment

Find outliers in the numerical variables (you might find outliers in almost all the variables), and check it using visual analysis

[45]:
```
Q1 = dd['actual_time'].quantile(0.25)
Q3 = dd['actual_time'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter out outliers
df_filtered = dd[(dd['actual_time'] >= lower_bound) & (dd['actual_time'] <=␣
 ↪upper_bound)]
df_filtered
```

[45]:
```
            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
3       training 2018-09-20 02:35:36.476840
4       training 2018-09-20 02:35:36.476840
…          …                          …
144862  training 2018-09-20 16:24:28.436231
144863  training 2018-09-20 16:24:28.436231
144864  training 2018-09-20 16:24:28.436231
144865  training 2018-09-20 16:24:28.436231
144866  training 2018-09-20 16:24:28.436231

                                    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
3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
…                                                     …        …
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    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)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
…                          …           …                          …
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)

       destination_center              destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
…                    …                            …
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                   od_start_time  …   destination_city  \
0      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
1      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
2      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
3      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
4      2018-09-20 03:21:32.418600  …  Khambhat_MotvdDPP
…                            …  …                …
144862 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144863 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144864 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144865 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur
144866 2018-09-20 16:24:28.436231  …   Gurgaon_Bilaspur

        destination_place_code  destination_state     source_city  \
0                            D            Gujarat   Anand_VUNagar
1                            D            Gujarat   Anand_VUNagar
2                            D            Gujarat   Anand_VUNagar
3                            D            Gujarat   Anand_VUNagar
4                            D            Gujarat   Anand_VUNagar
…                          …              …              …
144862                      HB            Haryana  Sonipat_Kundli
144863                      HB            Haryana  Sonipat_Kundli
144864                      HB            Haryana  Sonipat_Kundli
```

```
144865                     HB            Haryana  Sonipat_Kundli
144866                     HB            Haryana  Sonipat_Kundli

       source_place_code  source_state  Year  Month  Day  time_difference
0                     DC       Gujarat  2018      9   20        86.200000
1                     DC       Gujarat  2018      9   20        86.200000
2                     DC       Gujarat  2018      9   20        86.200000
3                     DC       Gujarat  2018      9   20        86.200000
4                     DC       Gujarat  2018      9   20        86.200000
…                    …            …    …  …              …
144862                 H       Haryana  2018      9   20       427.683333
144863                 H       Haryana  2018      9   20       427.683333
144864                 H       Haryana  2018      9   20       427.683333
144865                 H       Haryana  2018      9   20       427.683333
144866                 H       Haryana  2018      9   20       427.683333

[127809 rows x 34 columns]
```

for finding Outliers in other numerical columns and handling them

[46]:
```
import numpy as np

# Define a function to detect outliers using z-score
def detect_outliers_zscore(data, threshold=3):
    z_scores = np.abs((data - data.mean()) / data.std())
    return z_scores > threshold

# Define a function to detect outliers using interquartile range (IQR)
def detect_outliers_iqr(data):
    Q1 = np.percentile(data, 25)
    Q3 = np.percentile(data, 75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return (data < lower_bound) | (data > upper_bound)

# Columns to analyze for outliers
columns_to_check = ['actual_distance_to_destination', 'actual_time',␣
 ↪'osrm_time', 'osrm_distance', 'factor',
                    'segment_actual_time', 'segment_osrm_time',␣
 ↪'segment_osrm_distance', 'segment_factor', 'time_difference']

# Detect outliers using z-score method
outliers_zscore = dd[columns_to_check].apply(detect_outliers_zscore)

# Detect outliers using IQR method
outliers_iqr = dd[columns_to_check].apply(detect_outliers_iqr)
```

```
# Mark outliers in the DataFrame
dd['outlier_zscore'] = outliers_zscore.any(axis=1)
dd['outlier_iqr'] = outliers_iqr.any(axis=1)

# Handle outliers (e.g., replace with NaN)
# You can choose an appropriate method based on your data and analysis

# Replace outliers with NaN
dd[columns_to_check] = dd[columns_to_check].where(~(outliers_zscore |␣
 ↪outliers_iqr), np.nan)

# Print DataFrame after handling outliers
dd
```

[46]:

```
            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
3       training 2018-09-20 02:35:36.476840
4       training 2018-09-20 02:35:36.476840
…          …                          …
144862  training 2018-09-20 16:24:28.436231
144863  training 2018-09-20 16:24:28.436231
144864  training 2018-09-20 16:24:28.436231
144865  training 2018-09-20 16:24:28.436231
144866  training 2018-09-20 16:24:28.436231

                                    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
3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…    Carting
…                                                    …        …
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    Carting
144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…    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)
3       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
4       trip-153741093647649320  IND388121AAA  Anand_VUNagar_DC (Gujarat)
```

```
…                          …           …                          …
144862  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144863  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144864  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144865  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)
144866  trip-153746066843555182  IND131028AAB  Sonipat_Kundli_H (Haryana)

       destination_center              destination_name  \
0            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
1            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
2            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
3            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
4            IND388620AAB  Khambhat_MotvdDPP_D (Gujarat)
…                  …                             …
144862       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144863       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144864       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144865       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)
144866       IND000000ACB  Gurgaon_Bilaspur_HB (Haryana)

                   od_start_time  … destination_state     source_city  \
0      2018-09-20 03:21:32.418600  …           Gujarat   Anand_VUNagar
1      2018-09-20 03:21:32.418600  …           Gujarat   Anand_VUNagar
2      2018-09-20 03:21:32.418600  …           Gujarat   Anand_VUNagar
3      2018-09-20 03:21:32.418600  …           Gujarat   Anand_VUNagar
4      2018-09-20 03:21:32.418600  …           Gujarat   Anand_VUNagar
…                           …  …                …             …
144862 2018-09-20 16:24:28.436231  …           Haryana  Sonipat_Kundli
144863 2018-09-20 16:24:28.436231  …           Haryana  Sonipat_Kundli
144864 2018-09-20 16:24:28.436231  …           Haryana  Sonipat_Kundli
144865 2018-09-20 16:24:28.436231  …           Haryana  Sonipat_Kundli
144866 2018-09-20 16:24:28.436231  …           Haryana  Sonipat_Kundli

        source_place_code  source_state  Year  Month  Day  time_difference  \
0                      DC       Gujarat  2018      9   20        86.200000
1                      DC       Gujarat  2018      9   20        86.200000
2                      DC       Gujarat  2018      9   20        86.200000
3                      DC       Gujarat  2018      9   20        86.200000
4                      DC       Gujarat  2018      9   20        86.200000
…                    …            …    …    …  …               …
144862                  H       Haryana  2018      9   20       427.683333
144863                  H       Haryana  2018      9   20       427.683333
144864                  H       Haryana  2018      9   20       427.683333
144865                  H       Haryana  2018      9   20       427.683333
144866                  H       Haryana  2018      9   20       427.683333

        outlier_zscore  outlier_iqr
```

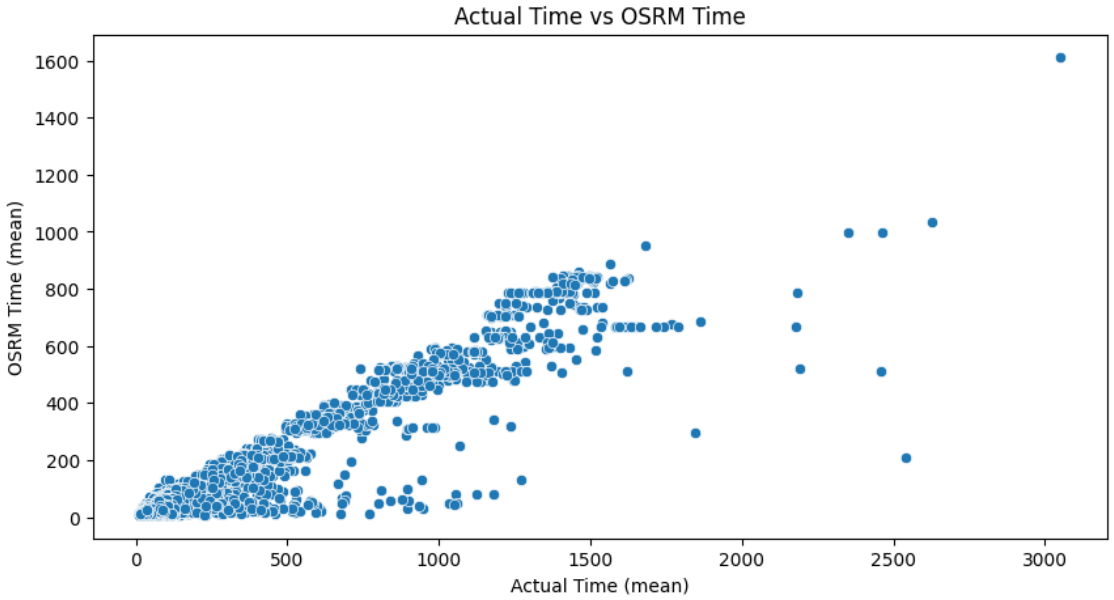
```
0                False        False
1                False        False
2                False        False
3                False        False
4                False        False
…                …          …
144862           False        False
144863           False        False
144864           False        False
144865           False        False
144866            True         True

[144316 rows x 36 columns]
```

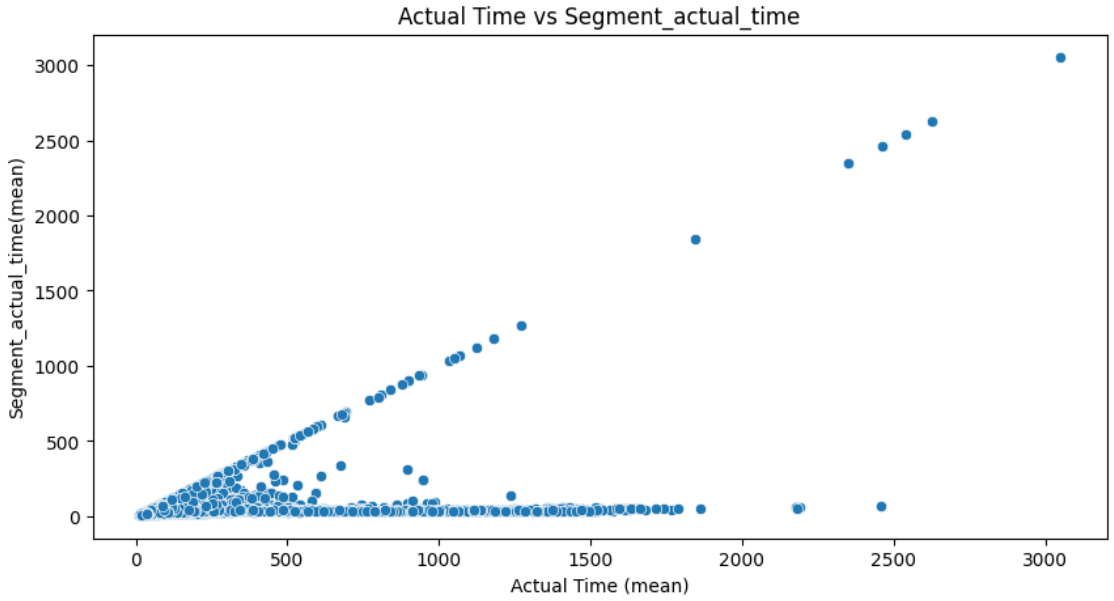
[47]:
```
df_clean_aggregate.to_csv('/content/final_data_agg.csv', index = False)
```

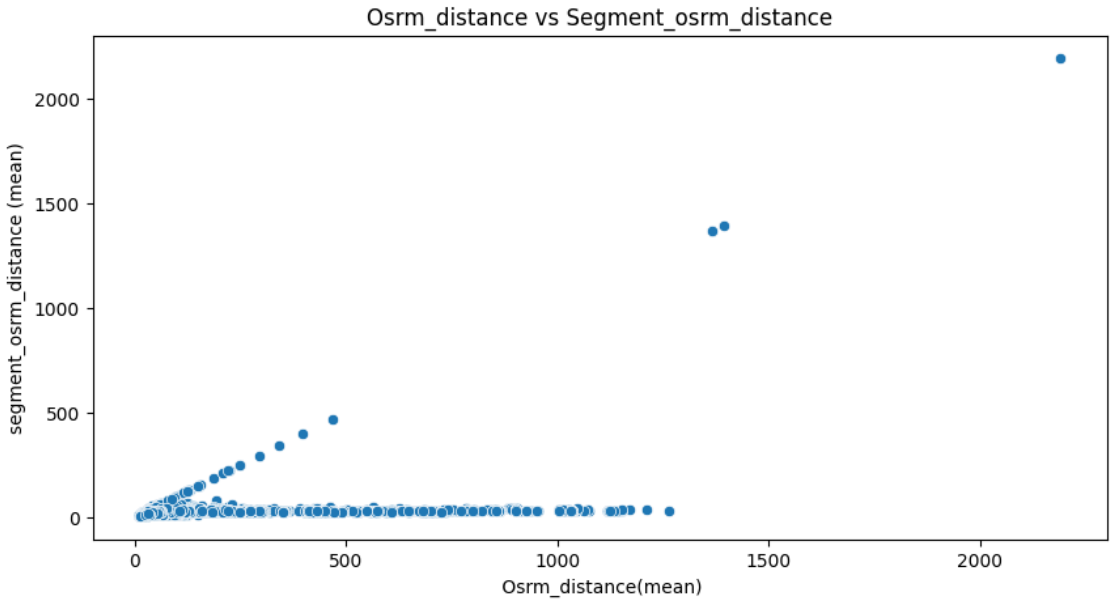
9 6. Checking relationship between aggregated fields

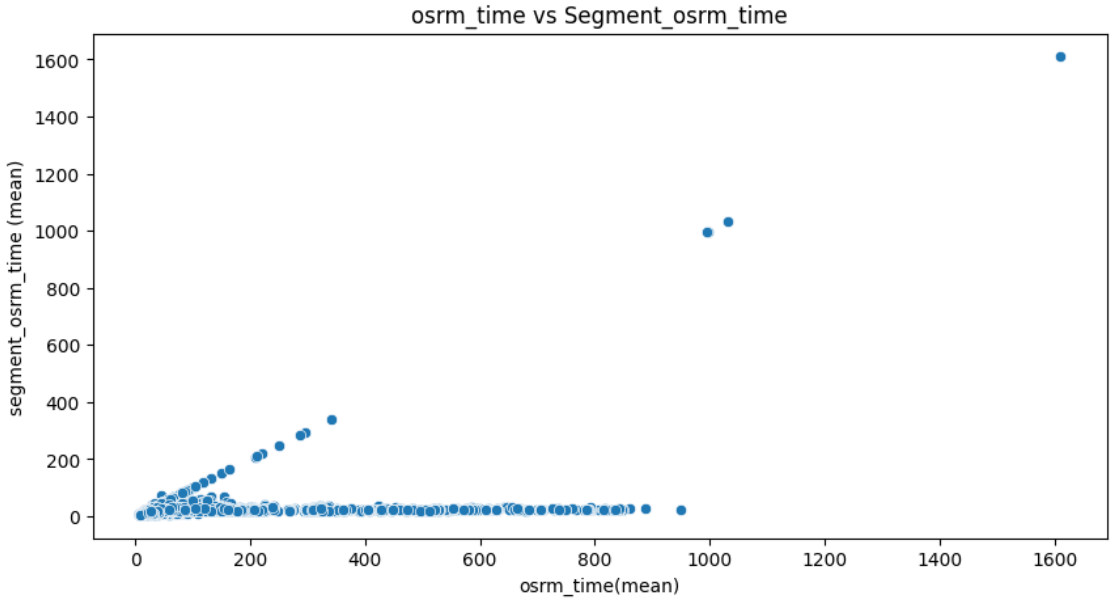
[48]:
```
plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['actual_time'], y=aggregated['osrm_time'])
plt.title('Actual Time vs OSRM Time')
plt.xlabel('Actual Time (mean)')
plt.ylabel('OSRM Time (mean)')
plt.show()


plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['actual_time'],␣
 ↪y=aggregated['segment_actual_time'])
plt.title('Actual Time vs Segment_actual_time')
plt.xlabel('Actual Time (mean)')
plt.ylabel('Segment_actual_time(mean)')
plt.show()


plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['osrm_distance'],␣
 ↪y=aggregated['segment_osrm_distance'])
plt.title('Osrm_distance vs Segment_osrm_distance')
plt.xlabel(' Osrm_distance(mean)')
plt.ylabel('segment_osrm_distance (mean)')
plt.show()


plt.figure(figsize=(10, 5))
sns.scatterplot(x=aggregated['osrm_time'], y=aggregated['segment_osrm_time'])
```

```
plt.title('osrm_time vs Segment_osrm_time')
plt.xlabel('osrm_time(mean)')
plt.ylabel('segment_osrm_time (mean)')
plt.show()
```









10 7. Handling categorical values (10 Points)

[49]:
```
from sklearn.preprocessing import LabelEncoder
df = dd
# Example of label encoding
encoder = LabelEncoder()
```

```
df['route_type_encoded'] = encoder.fit_transform(df['route_type'])
df[['route_type', 'route_type_encoded']].head()
```

[49]:

	route_type	route_type_encoded
0	Carting	0
1	Carting	0
2	Carting	0
3	Carting	0
4	Carting	0

[50]:
```
# One-hot encoding using pandas
df_one_hot = pd.get_dummies(df, columns=['route_type'], prefix='route')
df_one_hot.head()
```

[50]:

	data	trip_creation_time	route_schedule_uuid	trip_uuid	source_center	source_name	destination_center	destination_name	od_start_time	od_end_time	…	source_state	Year	Month	Day
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	…	Gujarat	2018	9	20
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	…	Gujarat	2018	9	20
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	…	Gujarat	2018	9	20
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	…	Gujarat	2018	9	20
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	2018-09-20 04:47:45.236797	…	Gujarat	2018	9	20

```
4 2018-09-20 04:47:45.236797  …       Gujarat  2018      9  20

   time_difference  outlier_zscore  outlier_iqr  route_type_encoded  \
0             86.2           False        False                   0
1             86.2           False        False                   0
2             86.2           False        False                   0
3             86.2           False        False                   0
4             86.2           False        False                   0

   route_Carting  route_FTL
0           True      False
1           True      False
2           True      False
3           True      False
4           True      False

[5 rows x 38 columns]
```

[50]:

11 8. Column Normalization /Column Standardization (10 Points)

Do one-hot encoding of categorical variables (like route_type) bold text

Normalize/ Standardize the numerical features using MinMaxScaler or StandardScaler.

[51]:
```
categorical_columns = ['route_type', 'source_name', 'destination_name']

# Performing one-hot encoding
df_encoded = pd.get_dummies(dd, columns=categorical_columns)

# Check the first few rows to verify
df_encoded.head()
```

[51]:
```
       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
3  training 2018-09-20 02:35:36.476840
4  training 2018-09-20 02:35:36.476840

                              route_schedule_uuid                trip_uuid  \
0  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…  trip-153741093647649320
1  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…  trip-153741093647649320
2  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…  trip-153741093647649320
```

```
3  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…  trip-153741093647649320
4  thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…  trip-153741093647649320

  source_center destination_center              od_start_time  \
0  IND388121AAA       IND388620AAB 2018-09-20 03:21:32.418600
1  IND388121AAA       IND388620AAB 2018-09-20 03:21:32.418600
2  IND388121AAA       IND388620AAB 2018-09-20 03:21:32.418600
3  IND388121AAA       IND388620AAB 2018-09-20 03:21:32.418600
4  IND388121AAA       IND388620AAB 2018-09-20 03:21:32.418600

                od_end_time  start_scan_to_end_scan  is_cutoff  …  \
0 2018-09-20 04:47:45.236797                    86.0       True  …
1 2018-09-20 04:47:45.236797                    86.0       True  …
2 2018-09-20 04:47:45.236797                    86.0       True  …
3 2018-09-20 04:47:45.236797                    86.0       True  …
4 2018-09-20 04:47:45.236797                    86.0      False  …

   destination_name_Wai_Central_DPP_3 (Maharashtra)  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

  destination_name_Wanaparthy_VallaDPP_D (Telangana)  \
0                                               False
1                                               False
2                                               False
3                                               False
4                                               False

   destination_name_Wankaner_JivanDPP_D (Gujarat)  \
0                                           False
1                                           False
2                                           False
3                                           False
4                                           False

   destination_name_Warangal_HunterRd_I (Telangana)  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

   destination_name_YamunaNagar_DC (Haryana)  \
0                                      False
```

```
1                                     False
2                                     False
3                                     False
4                                     False

   destination_name_Yavatmal_JajuDPP_D (Maharashtra)  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

   destination_name_Yellandu_Sudimala_D (Telangana)  \
0                                             False
1                                             False
2                                             False
3                                             False
4                                             False

   destination_name_Yellareddy_JKRoad_D (Telangana)  \
0                                             False
1                                             False
2                                             False
3                                             False
4                                             False

   destination_name_Zahirabad_Mohim_D (Telangana)  \
0                                            False
1                                            False
2                                            False
3                                            False
4                                            False

   destination_name_Zirakpur_DC (Punjab)
0                                  False
1                                  False
2                                  False
3                                  False
4                                  False

[5 rows x 2998 columns]
```

[53]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
numerical_features = df_encoded.select_dtypes(include=['float64', 'int64']).
 ↪columns
```

```
df_encoded[numerical_features] = scaler.
 ↪fit_transform(df_encoded[numerical_features])
df_encoded
```

[53]:
```
            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
3       training 2018-09-20 02:35:36.476840
4       training 2018-09-20 02:35:36.476840
…          …                          …
144862  training 2018-09-20 16:24:28.436231
144863  training 2018-09-20 16:24:28.436231
144864  training 2018-09-20 16:24:28.436231
144865  training 2018-09-20 16:24:28.436231
144866  training 2018-09-20 16:24:28.436231

                                    route_schedule_uuid  \
0       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…
1       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…
2       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…
3       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…
4       thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3…
…                                                     …
144862  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…
144863  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…
144864  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…
144865  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…
144866  thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5…

                     trip_uuid source_center destination_center  \
0       trip-153741093647649320  IND388121AAA       IND388620AAB
1       trip-153741093647649320  IND388121AAA       IND388620AAB
2       trip-153741093647649320  IND388121AAA       IND388620AAB
3       trip-153741093647649320  IND388121AAA       IND388620AAB
4       trip-153741093647649320  IND388121AAA       IND388620AAB
…                           …           …                  …
144862  trip-153746066843555182  IND131028AAB       IND000000ACB
144863  trip-153746066843555182  IND131028AAB       IND000000ACB
144864  trip-153746066843555182  IND131028AAB       IND000000ACB
144865  trip-153746066843555182  IND131028AAB       IND000000ACB
144866  trip-153746066843555182  IND131028AAB       IND000000ACB

                   od_start_time               od_end_time  \
0      2018-09-20 03:21:32.418600 2018-09-20 04:47:45.236797
1      2018-09-20 03:21:32.418600 2018-09-20 04:47:45.236797
2      2018-09-20 03:21:32.418600 2018-09-20 04:47:45.236797
```

```
3      2018-09-20 03:21:32.418600 2018-09-20 04:47:45.236797
4      2018-09-20 03:21:32.418600 2018-09-20 04:47:45.236797
…                             …                          …
144862 2018-09-20 16:24:28.436231 2018-09-20 23:32:09.618069
144863 2018-09-20 16:24:28.436231 2018-09-20 23:32:09.618069
144864 2018-09-20 16:24:28.436231 2018-09-20 23:32:09.618069
144865 2018-09-20 16:24:28.436231 2018-09-20 23:32:09.618069
144866 2018-09-20 16:24:28.436231 2018-09-20 23:32:09.618069

        start_scan_to_end_scan  is_cutoff  …  \
0                    -0.845502       True  …
1                    -0.845502       True  …
2                    -0.845502       True  …
3                    -0.845502       True  …
4                    -0.845502      False  …
…                        …         …  …
144862               -0.517010       True  …
144863               -0.517010       True  …
144864               -0.517010       True  …
144865               -0.517010       True  …
144866               -0.517010      False  …

        destination_name_Wai_Central_DPP_3 (Maharashtra)  \
0                                                  False
1                                                  False
2                                                  False
3                                                  False
4                                                  False
…                                                      …
144862                                             False
144863                                             False
144864                                             False
144865                                             False
144866                                             False

       destination_name_Wanaparthy_VallaDPP_D (Telangana)  \
0                                                   False
1                                                   False
2                                                   False
3                                                   False
4                                                   False
…                                                       …
144862                                              False
144863                                              False
144864                                              False
144865                                              False
144866                                              False
```

```
        destination_name_Wankaner_JivanDPP_D (Gujarat)  \
0                                                False
1                                                False
2                                                False
3                                                False
4                                                False
…                                                …
144862                                           False
144863                                           False
144864                                           False
144865                                           False
144866                                           False

        destination_name_Warangal_HunterRd_I (Telangana)  \
0                                                  False
1                                                  False
2                                                  False
3                                                  False
4                                                  False
…                                                  …
144862                                             False
144863                                             False
144864                                             False
144865                                             False
144866                                             False

        destination_name_YamunaNagar_DC (Haryana)  \
0                                           False
1                                           False
2                                           False
3                                           False
4                                           False
…                                           …
144862                                      False
144863                                      False
144864                                      False
144865                                      False
144866                                      False

        destination_name_Yavatmal_JajuDPP_D (Maharashtra)  \
0                                                   False
1                                                   False
2                                                   False
3                                                   False
4                                                   False
…                                                   …
```

```
144862                                                False
144863                                                False
144864                                                False
144865                                                False
144866                                                False

        destination_name_Yellandu_Sudimala_D (Telangana)  \
0                                                     False
1                                                     False
2                                                     False
3                                                     False
4                                                     False
…                                                     …
144862                                                False
144863                                                False
144864                                                False
144865                                                False
144866                                                False

        destination_name_Yellareddy_JKRoad_D (Telangana)  \
0                                                     False
1                                                     False
2                                                     False
3                                                     False
4                                                     False
…                                                     …
144862                                                False
144863                                                False
144864                                                False
144865                                                False
144866                                                False

        destination_name_Zahirabad_Mohim_D (Telangana)  \
0                                                   False
1                                                   False
2                                                   False
3                                                   False
4                                                   False
…                                                   …
144862                                              False
144863                                              False
144864                                              False
144865                                              False
144866                                              False

        destination_name_Zirakpur_DC (Punjab)
0                                       False
```

```
1                                       False
2                                       False
3                                       False
4                                       False
…                                       …
144862                                  False
144863                                  False
144864                                  False
144865                                  False
144866                                  False

[144316 rows x 2998 columns]
```

12 9. Business Insights (10 Points) -

Should include patterns observed in the data along with what you can infer from it. Eg: Check from where most orders are coming from (State, Corridor etc) Busiest corridor, avg distance between them, avg time taken

[54]:
```
df=dd
# Most common source states or cities
most_common_sources = df['source_state'].value_counts().head()
print("Most Common Source States:\n", most_common_sources)

# Most common destination states or cities
most_common_destinations = df['destination_state'].value_counts().head()
print("Most Common Destination States:\n", most_common_destinations)

# Identifying the busiest corridors
busiest_corridors = df.groupby(['source_state', 'destination_state']).size().
 ↪nlargest(1)
print("Busiest Corridor:\n", busiest_corridors)
```

```
Most Common Source States:
 source_state
Haryana        27408
Maharashtra    21401
Karnataka      19562
Tamil Nadu      7494
Gujarat         7202
Name: count, dtype: int64
Most Common Destination States:
 destination_state
Karnataka      21049
Haryana        20602
Maharashtra    18196
West Bengal     8499
```

```
Telangana       8200
Name: count, dtype: int64
Busiest Corridor:
 source_state  destination_state
Maharashtra   Maharashtra          11876
dtype: int64
```

[55]:
```
# Calculating average distance and time for the busiest corridor
if not busiest_corridors.empty:
    busiest_route = busiest_corridors.index[0]
    avg_metrics = df[(df['source_state'] == busiest_route[0]) &␣
 ↪(df['destination_state'] == busiest_route[1])].agg({
        'actual_distance_to_destination': 'mean',
        'actual_time': 'mean'
    })
    print(f"Average Metrics for Busiest Corridor {busiest_route}:\n",␣
 ↪avg_metrics)
```

```
Average Metrics for Busiest Corridor ('Maharashtra', 'Maharashtra'):
 actual_distance_to_destination     64.032019
actual_time                        137.225921
dtype: float64
```

In general, the key Business Insights are:

Route Efficiency Variability: Significant differences between actual delivery times and OSRM estimated times suggest inefficiencies in some routes, possibly due to traffic or suboptimal routing.

Demand Concentration: High trip frequencies in specific states or cities highlight major demand hubs, suggesting the need for enhanced resource allocation in these areas.

Temporal Demand Patterns: Clear patterns in trip frequencies across months and weekdays indicate seasonal and weekly demand fluctuations, which can guide resource scheduling and capacity planning.

Transport Mode Effectiveness: Differences in efficiency between FTL and Carting modes may indicate that one mode could be better suited for certain types of deliveries or routes.

Emerging Markets Identification: Increasing logistical activities in previously lower-demand areas suggest potential emerging markets, offering opportunities for strategic expansion.

[55]:

13 10. Recommendations (10 Points) -

Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand.

Based on the insights, here are some actionable recommendations:

1. Expand Capacity: For the busiest corridors, especially where delays or long transit times are observed, consider increasing logistics capacity—more trucks, better route management.Most common source and destination are Karnataka, Haryana, and Maharashtra. The inter and intra state capacity must be increased for these states.
2. Optimize Routes: If certain routes show inefficiencies, investigate alternative routes or times to bypass heavy traffic or delays.

based on the data:

Average Metrics for Busiest Corridor ('Maharashtra', 'Maharashtra'): actual_distance_to_destination 64.032019 actual_time 137.225921

Expressways must be built for such busiest corridor.

3. Focus on Top Sources: For areas generating most orders, ensure resources such as packaging, handling, and dispatch are adequately scaled to meet demand. Top sources are Haryana, Maharashtra and Karnataka.

In General,

1. Optimize High Traffic Routes Action Item: Identify and prioritize the optimization of the busiest corridors, such as the routes between major states or cities that have been identified as having the highest volume of deliveries. Implementation: Use the route feature to concentrate resources, enhance scheduling precision, and allocate more fleet assets to these corridors. Consider alternate routes or off-peak delivery schedules to alleviate congestion and reduce delivery times.
2. Enhance Time and Distance Efficiency Action Item: Address routes where the time_efficiency and distance_efficiency metrics indicate significant discrepancies between the actual and OSRM-projected values. Implementation: Investigate the causes of these inefficiencies—whether they stem from traffic, suboptimal routing, or other logistical challenges. Implement more dynamic routing algorithms that can adapt to real-time conditions such as traffic or weather.
3. Resource Allocation Based on Demand Forecasting Action Item: Leverage the extracted date and time features (like trip_creation_weekday and trip_creation_hour) to forecast peak demand periods and allocate resources accordingly. Implementation: Develop predictive models that utilize historical delivery data to forecast demand surges. Adjust staffing levels, vehicle availability, and warehouse operations based on these forecasts to manage workload efficiently during peak and off-peak hours.
4. Focus on Geographical Expansion Action Item: Expand services in regions where source_state and destination_state analysis shows growing demand but current service levels are low. Implementation: Analyze the geographical data to identify potential new markets or underserved areas. Plan logistics infrastructure development in these areas, such as setting up new warehouses or partnering with local businesses for quicker dispatch and delivery.
5. Improve Data Accuracy and Collection Action Item: Enhance the accuracy of the data collected at every point in the delivery process to improve the reliability of forecasting and operational planning. Implementation: Implement robust data validation rules in the data collection process to ensure accuracy. Regularly train staff on the importance of accurate data entry and use automated systems to reduce human error.

6. Customer-Centric Delivery Options Action Item: Offer more flexible delivery options to enhance customer satisfaction, particularly in busy corridors or for deliveries that consistently experience delays. Implementation: Provide customers with options for delivery times and update them in real-time on their delivery status. Consider implementing last-mile delivery innovations such as localized delivery hubs or partnerships with local retailers.

[56]: