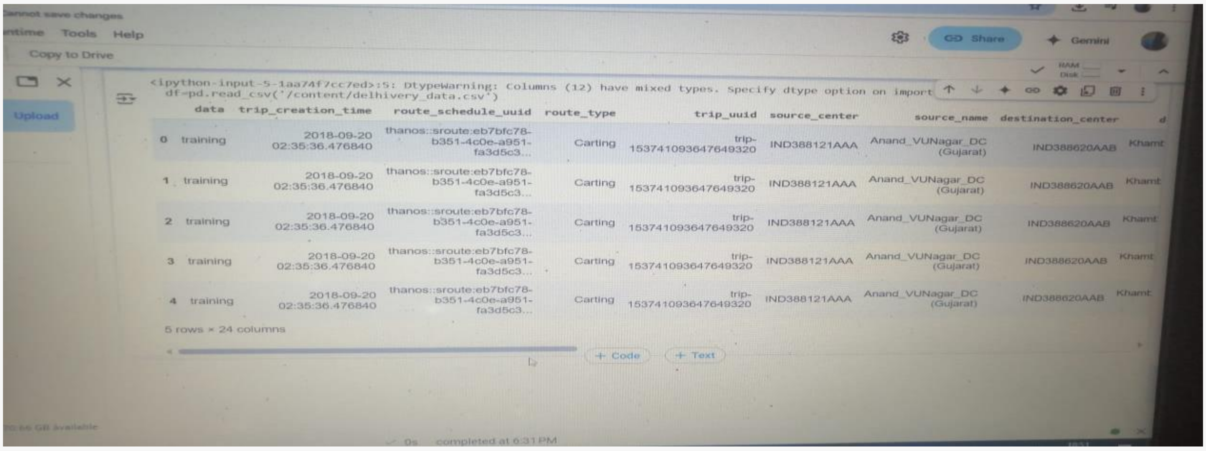
DELHIVERY CASE STUDY

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('/content/delhivery_data.csv')
df.head()
```



```
<ipython-input-5-1aa74f7cc7ed>:5: DtypeWarning: Columns (12) have mixed types. Specify dtype option on import
  df=pd.read_csv('/content/delhivery_data.csv')
      data  trip_creation_time  route_schedule_uuid  route_type              trip_uuid  source_center          source_name  destination_center  d
0 training          2018-09-20  thanos::sroute:eb7bfc78-  Carting                 trip-  IND388121AAA  Anand_VUNagar_DC       IND388620AAB  Khamt
           02:35:36.476840         b351-4c0e-a951-            153741093647649320                   (Gujarat)
                                       fa3d5c3...
1 training          2018-09-20  thanos::sroute:eb7bfc78-  Carting                 trip-  IND388121AAA  Anand_VUNagar_DC       IND388620AAB  Khamt
           02:35:36.476840         b351-4c0e-a951-            153741093647649320                   (Gujarat)
                                       fa3d5c3...
2 training          2018-09-20  thanos::sroute:eb7bfc78-  Carting                 trip-  IND388121AAA  Anand_VUNagar_DC       IND388620AAB  Khamt
           02:35:36.476840         b351-4c0e-a951-            153741093647649320                   (Gujarat)
                                       fa3d5c3...
3 training          2018-09-20  thanos::sroute:eb7bfc78-  Carting                 trip-  IND388121AAA  Anand_VUNagar_DC       IND388620AAB  Khamt
           02:35:36.476840         b351-4c0e-a951-            153741093647649320                   (Gujarat)
                                       fa3d5c3...
4 training          2018-09-20  thanos::sroute:eb7bfc78-  Carting                 trip-  IND388121AAA  Anand_VUNagar_DC       IND388620AAB  Khamt
           02:35:36.476840         b351-4c0e-a951-            153741093647649320                   (Gujarat)
                                       fa3d5c3...
5 rows × 24 columns
```

Looking at the data we can work on few questions to improve the delhivery business case

Questions for Business Improvement

These questions are categorized based on the key areas of focus for improvement – Delivery Performance, Geographic Operations, Customer Experience, and Operational Efficiency:

1. Delivery Performance:

- **What are the average delivery times for different regions and product categories?**
- **What is the variability in delivery times for specific routes or delivery partners?**
- **What is the on-time delivery rate for different delivery windows and service types? What are the primary reasons for delivery failures and returns? How do external factors like weather or traffic conditions impact delivery performance?**
- **Are there specific delivery partners or personnel with consistently higher or lower performance?**

2. Geographic Operations:

- **Which geographic areas have the highest delivery density and volume?**
- **Are there regions with consistently longer delivery times or higher failure rates?**
- **Are there opportunities to expand service to new geographic areas with high demand?**
- **How does the infrastructure and accessibility in different regions affect delivery performance?**
- **What are the most popular product categories and their geographic distribution?**

- **What are customer preferences for delivery speed, time windows, and other delivery options?**
- **How satisfied are customers with the overall delivery experience?**
- **Are there any bottlenecks in the delivery process that are causing delays or failures?**
- **Can we leverage technology or automation to improve delivery efficiency?**
- **How can we better predict and manage peak delivery periods?**
- **Are there opportunities to improve communication and coordination between different teams involved in the delivery process?**
- **Can we reduce operational costs without compromising delivery quality or speed?**

3. Customer Experience:

- **What are the most popular product categories and their geographic distribution?** (Tailor inventory and marketing)
- **What are customer preferences for delivery speed, time windows, and other delivery options?** (Offer customized services)
- **How satisfied are customers with the overall delivery experience?** (Measure customer feedback and sentiment)
- **Are there any recurring customer complaints or issues related to deliveries?** (Address pain points and improve satisfaction)
- **Can we implement personalized delivery options based on customer history and preferences?** (Enhance customer loyalty)

4. Operational Efficiency:

- **Are there any bottlenecks in the delivery process that are causing delays or failures?** (Optimize workflows)
- **Can we leverage technology or automation to improve delivery efficiency?** (Streamline operations)
- **How can we better predict and manage peak delivery periods?** (Ensure resource availability)
- **Are there opportunities to improve communication and coordination between different teams involved in the delivery process?** (Enhance collaboration)
- **Can we reduce operational costs without compromising delivery quality or speed?** (Identify cost-saving measures

```
df.info()
#   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   data                            111981 non-null  object
 1   trip_creation_time              111981 non-null  object
 2   route_schedule_uuid             111981 non-null  object
 3   route_type                      111981 non-null  object
 4   trip_uuid                       111981 non-null  object
 5   source_center                   111981 non-null  object
 6   source_name                     111771 non-null  object
 7   destination_center              111981 non-null  object
 8   destination_name                111824 non-null  object
 9   od_start_time                   111981 non-null  object
 10  od_end_time                     111981 non-null  object
 11  start_scan_to_end_scan          111980 non-null  float64
 12  is_cutoff                       111980 non-null  object
 13  cutoff_factor                   111980 non-null  float64
 14  cutoff_timestamp                111980 non-null  object
 15  actual_distance_to_destination  111980 non-null  float64
 16  actual_time                     111980 non-null  float64
 17  osrm_time                       111980 non-null  float64
 18  osrm_distance                   111980 non-null  float64
 19  factor                          111980 non-null  float64
 20  segment_actual_time             111980 non-null  float64
 21  segment_osrm_time               111980 non-null  float64
 22  segment_osrm_distance           111980 non-null  float64

    23  segment_factor                  111980 non-null  float64
```

So here we have 24 columns and 111981 rows out of which some of the columns are having only one missing value, which could be filled by their mean

Only source_name and destination_name are having more null values and which could not be filled by anything . so we will try to remove the rows in which source_name and destination_name are missing

```
df = df[~df['source_name'].isnull() & ~df['destination_name'].isnull()]
```

Now we have the data which all the rows in which source_name and destination_name were missing are now filtered

```
numeric_cols = df.select_dtypes(include=np.number).columns
for col in numeric_cols:
    df[col] = df[col].fillna(df[col].mean())
```

By running the above code the missing values will be filled by the mean of the column in which they are present

```
df.groupby('source_center').size().sort_values(ascending=False)
```

```
source_center

IND000000ACB    17883

IND562132AAA     8054

IND421302AAG     7085

IND411033AAA     3225

IND501359AAE     2591
```

This are the top 5 source center

```python
df.groupby('destination_center').size().sort_values(ascending=False)
```

```
                              destination_center
IND000000ACB                              11748
IND562132AAA                               8100
IND421302AAG                               4298
IND501359AAE                               3899
IND712311AAA                               3784
```

This are the top 5 active destination center

IMPORTANT OUTCOME: THE INFRASTRUCTURE AND CAPACITY OF THESE ACTIVE SOURCE AND DESTINATION CENTRES SHOULD BE INCREASE WHICH WILL LEAD TO BETTER CUSTOMER SATISFACTION AND MORE ORDERS AND MORE PROFIT FOR BUSINESS.

```python
df['time_diff'] = df['actual_time'] - df['osrm_time']
grouped_data = df.groupby('destination_center')['time_diff'].mean()
grouped_data.sort_values(ascending=False)
```

```
IND490023AAA                              818.000
IND712311AAA                              573.699
IND110037AAM                              547.412
IND416510AAA                              510.925
IND302014AAA                              490.674
     ...
IND792001AAA                              -26.187
IND243301AAB                              -27.375
IND624101AAA                              -30.280
IND345001AAA                              -31.738
```

So we have destination centres where the
time difference is maximum

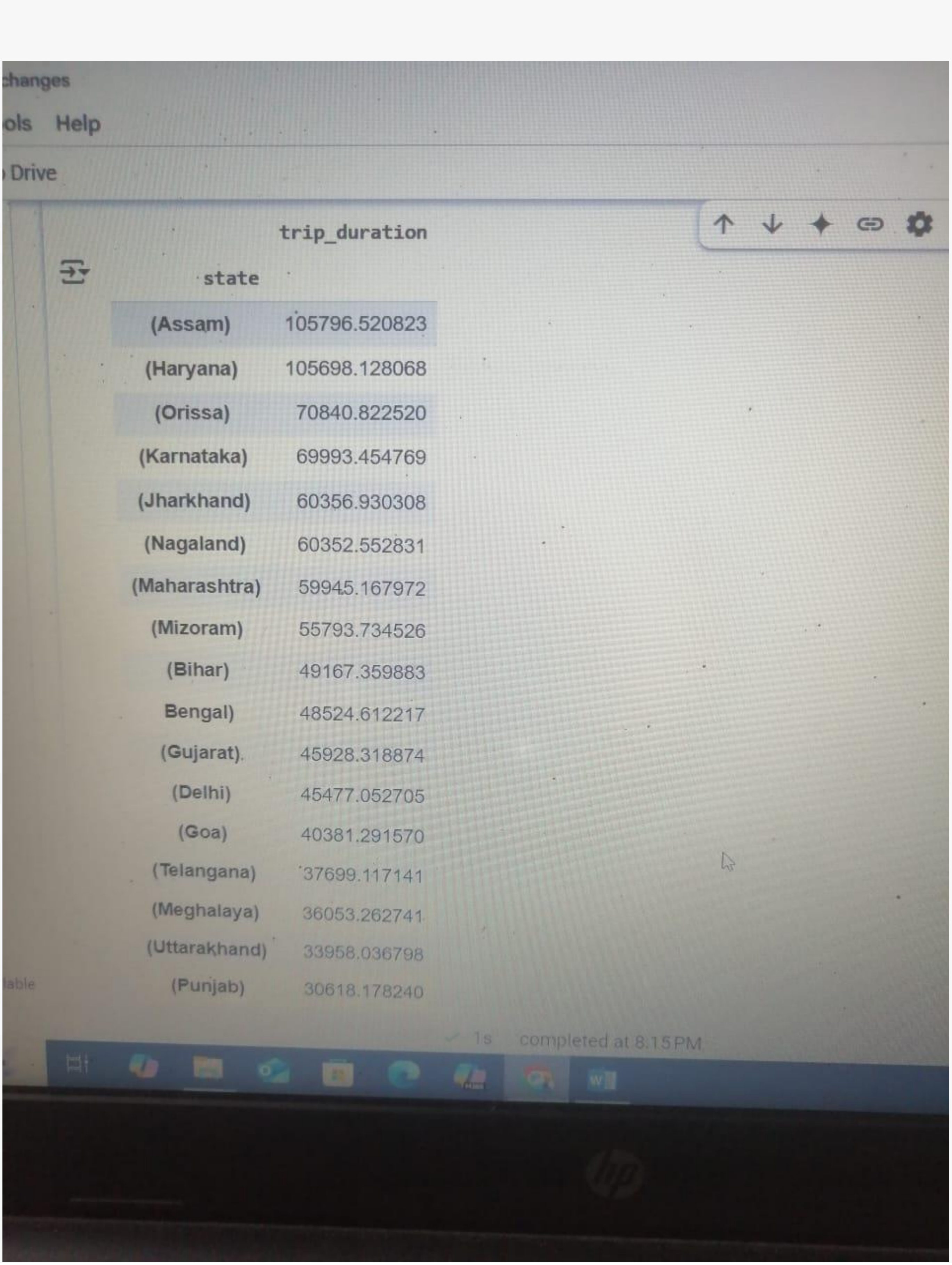
- Calculate the time taken between od_start_time and od_end_time and keep it as a feature. Drop the original columns, if required

```
df['od_end_time'] = pd.to_datetime(df['od_end_time'], format='%Y-%m-%d
%H:%M:%S.%f', errors='coerce')
df['od_end_time'] =
df['od_end_time'].fillna(pd.to_datetime(df['od_end_time'].astype(str),
errors='coerce'))
df['trip_duration'] = (df['od_end_time'] -
df['od_start_time']).dt.total_seconds()

a=df['trip_duration'].mean()
print(a)
```

```
57609.33816088323
On an average the difference between od end time and od start time is 57610 seconds
```

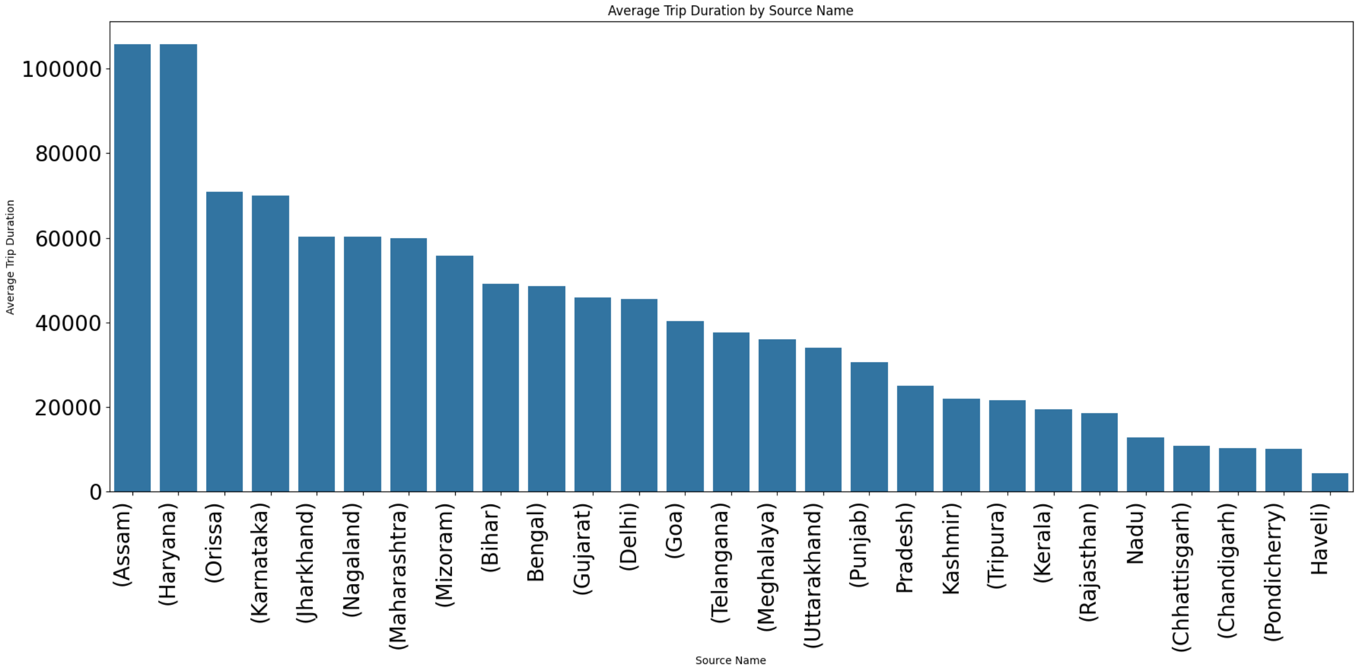
```
df['state'] = df['source_name'].str.split().str[-1]
df['state']
```

```
df.groupby('state')['trip_duration'].mean().sort_values(ascending=False)
```



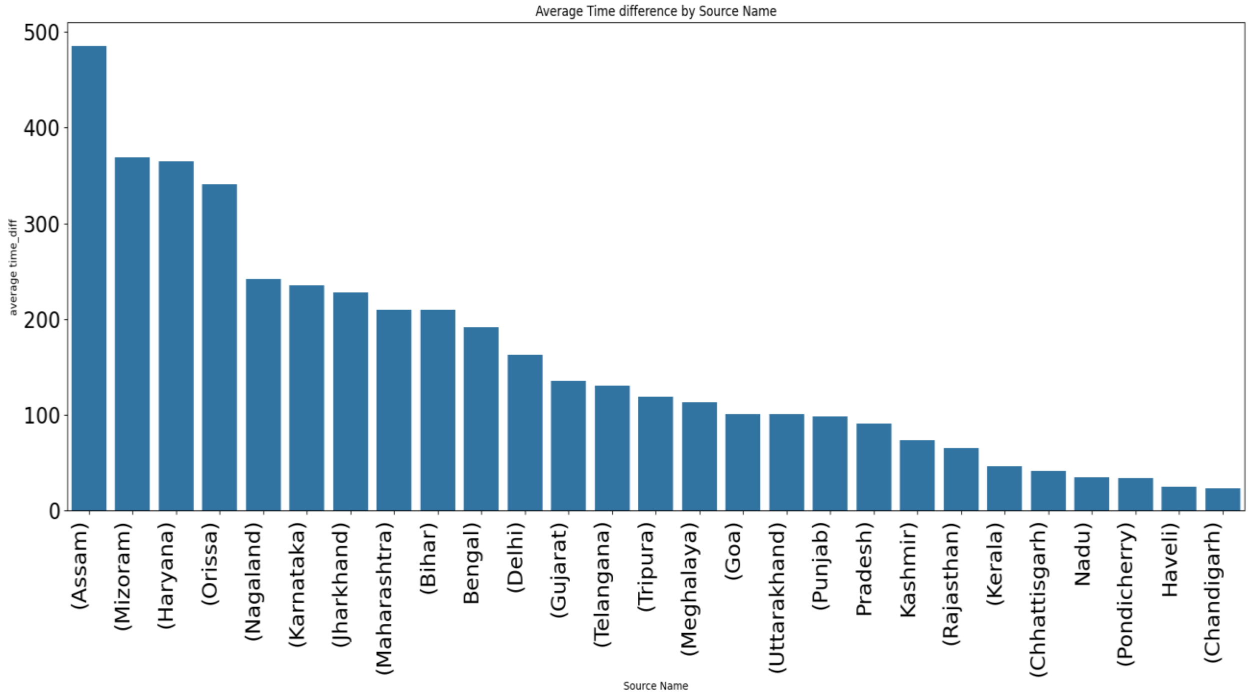
```
                 trip_duration
         state
       (Assam)   105796.520823
     (Haryana)   105698.128068
      (Orissa)    70840.822520
   (Karnataka)    69993.454769
   (Jharkhand)    60356.930308
    (Nagaland)    60352.552831
 (Maharashtra)    59945.167972
     (Mizoram)    55793.734526
       (Bihar)    49167.359883
       Bengal)    48524.612217
     (Gujarat)    45928.318874
       (Delhi)    45477.052705
         (Goa)    40381.291570
   (Telangana)    37699.117141
   (Meghalaya)    36053.262741
 (Uttarakhand)    33958.036798
      (Punjab)    30618.178240
```

```
a=df.groupby('state')['trip_duration'].mean().sort_values(ascending=False)
plt.figure(figsize=(18, 9))
sns.barplot(x=a.index, y=a.values)
plt.xlabel("Source Name")
plt.ylabel("Average Trip Duration")
plt.title("Average Trip Duration by Source Name")
plt.xticks(rotation=90, ha='right',fontsize=20)
plt.yticks(fontsize=20)

plt.tight_layout()
plt.show()
```



This is the graph showing the average trip duration coming out of different states. The states showing less average trip duration like Kashmir,Tripura,Kerala,Rajasthan etc. are matter of discussions

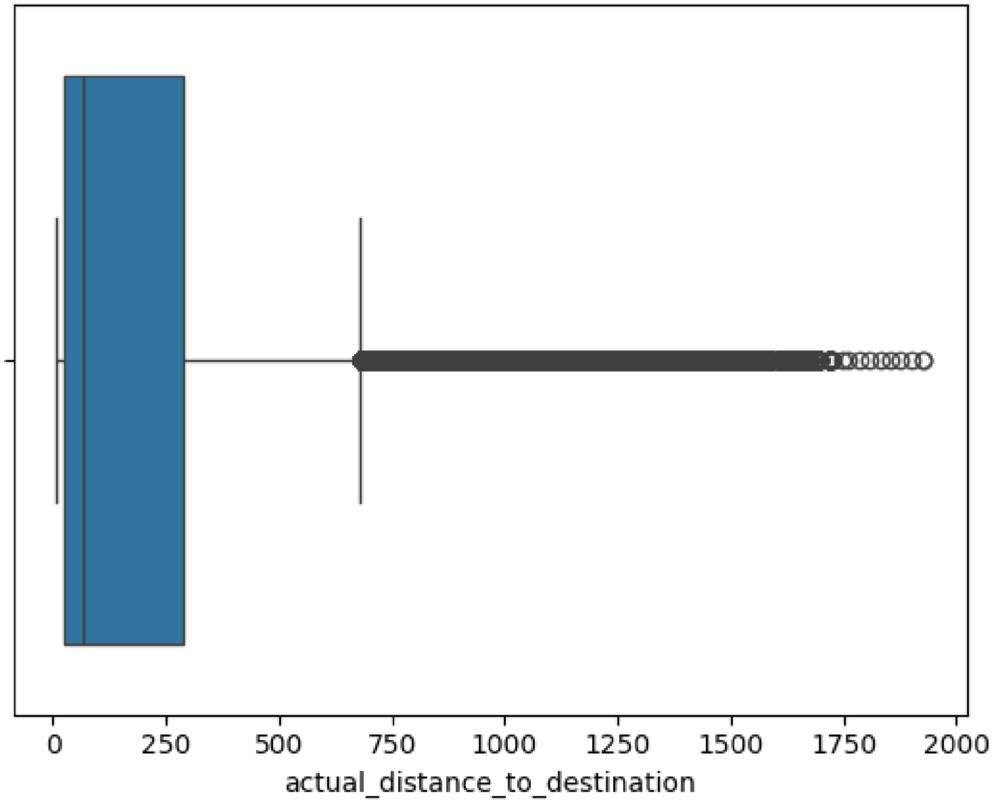
```
a=df.groupby('state')['time_diff'].mean().sort_values(ascending=False)
plt.figure(figsize=(18, 9))
sns.barplot(x=a.index, y=a.values)
plt.xlabel("Source Name")
plt.ylabel("average time_diff")
plt.title("Average Time difference by Source Name")
plt.xticks(rotation=90, ha='right',fontsize=20)
plt.yticks(fontsize=20)

plt.tight_layout()
plt.show()
```



This are the states with time difference between actual_time and osrm_time.

So states with higher time difference should have excellent communication and transportation to reduce this difference which leads to customer satisfaction and better profitability.

```
sns.boxplot(x=df['actual_distance_to_destination'])
```



So the actual distance to destination is mostly between 10 to 260 km. but there are many outliers which lies between 725 and 1800 km

Working on route is very important to reduce the actual distance . teams should work on this