

```
import pandas as pd
```

```
data =pd.read_csv(r'https://d2beiqkhq929f8.cloudfront.net/public_assets/assets/000/001/551/original/delhiery_data.csv?1642751181')
```

```
data
```

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_center	source_name	destination_center	destination_name	od_start_time	...	cutoff_timestamp	actual_distance_to_d
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	...	2018-09-20 04:27:55	
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	...	2018-09-20 04:17:55	
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	...	2018-09-20 04:01:19.505586	
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	...	2018-09-20 03:39:57	
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND388121AAA	Anand_VUNagar_DC (Gujarat)	IND388620AAB	Khambhat_MotvdDPP_D (Gujarat)	2018-09-20 03:21:32.418600	...	2018-09-20 03:33:55	
...
144862	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000000ACB	Gurgaon_Bilaspur_HB (Haryana)	2018-09-20 16:24:28.436231	...	2018-09-20 21:57:20	
144863	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000000ACB	Gurgaon_Bilaspur_HB (Haryana)	2018-09-20 16:24:28.436231	...	2018-09-20 21:31:18	
144864	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000000ACB	Gurgaon_Bilaspur_HB (Haryana)	2018-09-20 16:24:28.436231	...	2018-09-20 21:11:18	
144865	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000000ACB	Gurgaon_Bilaspur_HB (Haryana)	2018-09-20 16:24:28.436231	...	2018-09-20 20:53:19	
144866	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND131028AAB	Sonipat_Kundli_H (Haryana)	IND000000ACB	Gurgaon_Bilaspur_HB (Haryana)	2018-09-20 16:24:28.436231	...	2018-09-20 16:24:28.436231	

144867 rows × 24 columns

```
data.isnull().sum()
```

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	

```
data=data.dropna()
```

```
data.isnull().sum()
```

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	

```
data.duplicated().any()
```

False

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

```
<ipython-Input-105-22e118d719e9>:1: 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
data['diff']=pd.to_datetime(data['od_end_time'])-pd.to_datetime(data['od_start_time'])
```

```
data.head()
# we are going to encode categorical values to numerical values because ml models cant understand categorical values
# so we are going to convert
# in order to convert categorical values to numerical values i am using one hot encoder
from sklearn.preprocessing import LabelEncoder
import numpy as np
enc=LabelEncoder()

data['data']=enc.fit_transform(data['data'])
data['route_type']=enc.fit_transform(data['route_type'])
data['source_center']=enc.fit_transform(data['source_center'])
data['source_name']=enc.fit_transform(data['source_name'])
data['destination_center']=enc.fit_transform(data['destination_center'])
data['destination_name']=enc.fit_transform(data['destination_name'])
data['diff']=enc.fit_transform(data['diff'])

<ipython-Input-106-bc988f013baa>:9: 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
data['data']=enc.fit_transform(data['data'])
<ipython-Input-106-bc988f013baa>:10: 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
data['route_type']=enc.fit_transform(data['route_type'])
<ipython-Input-106-bc988f013baa>:11: 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
data['source_center']=enc.fit_transform(data['source_center'])
<ipython-Input-106-bc988f013baa>:12: 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
data['source_name']=enc.fit_transform(data['source_name'])
<ipython-Input-106-bc988f013baa>:13: 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
data['destination_center']=enc.fit_transform(data['destination_center'])
<ipython-Input-106-bc988f013baa>:14: 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
data['destination_name']=enc.fit_transform(data['destination_name'])
<ipython-Input-106-bc988f013baa>:15: 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
data['diff']=enc.fit_transform(data['diff'])
```

data

	data	trip_creation_time	route_schedule_uid	route_type	trip_uuid	source_center	source_name	destination_center	destination_name	od_start_time	...	actual_distance_to_destination	actual_time	osrm_t
0	1	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	0	153741093647649320	492	48	486	739	2018-09-20 03:21:32.418600	...	10.435660	14.0	1
1	1	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	0	153741093647649320	492	48	486	739	2018-09-20 03:21:32.418600	...	18.936842	24.0	2
2	1	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	0	153741093647649320	492	48	486	739	2018-09-20 03:21:32.418600	...	27.637279	40.0	2
3	1	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	0	153741093647649320	492	48	486	739	2018-09-20 03:21:32.418600	...	36.118028	62.0	4
4	1	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	0	153741093647649320	492	48	486	739	2018-09-20 03:21:32.418600	...	39.386040	68.0	4
...
144862	1	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	0	153746066843555182	76	1351	6	520	2018-09-20 16:24:28.436231	...	45.258278	94.0	€
144863	1	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	0	153746066843555182	76	1351	6	520	2018-09-20 16:24:28.436231	...	54.092531	120.0	7
144864	1	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	0	153746066843555182	76	1351	6	520	2018-09-20 16:24:28.436231	...	66.163591	140.0	€
144865	1	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	0	153746066843555182	76	1351	6	520	2018-09-20 16:24:28.436231	...	73.680667	158.0	€
144866	1	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	0	153746066843555182	76	1351	6	520	2018-09-20 16:24:28.436231	...	70.039010	426.0	€

144316 rows × 25 columns

Start coding on [generate](#) with AI.

```
# standardization of data in ml some of ml models will predict inaccurately because they are giving more importance to the
# higher values for lower values they are providing less importance while predicting values for that feature scaling is required in ml prospective
# here i am opting robustscaler it is a standardization technique in order to convert bigger values and lower values to particular range like 0 to 1 or -1 to +1 like that
from sklearn.preprocessing import StandardScaler
numeric_cols=data.select_dtypes(include='number')
ec=StandardScaler()

trans=ec.fit_transform(numeric_cols)
trans
```

```
array([[ 0.61585073, -1.48120305, -0.24782905, ..., -0.6074951 ,
        -0.19485494, -1.73031872],
       [ 0.61585073, -1.48120305, -0.24782905, ..., -0.73098455,
        -0.228145 , -1.73031872],
       [ 0.61585073, -1.48120305, -0.24782905, ..., -0.67186768,
        0.0138024 , -1.73031872],
       ...,
       [ 0.61585073, -1.48120305, -1.16872494, ..., -0.11830609,
        -0.33584814, 0.35975088],
       [ 0.61585073, -1.48120305, -1.16872494, ..., -0.21999471,
        -0.32732164, 0.35975088],
       [ 0.61585073, -1.48120305, -1.16872494, ..., -0.78416846,
        5.67667956, 0.35975088]])
```

```
trans=pd.DataFrame(trans,columns=numeric_cols.columns)
```

trans

	data	route_type	source_center	source_name	destination_center	destination_name	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	actual_time	osrm_time	osrm_distance	factor	segment_ac
0	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.650441	-0.649164	-0.674521	-0.659552	-0.648741	-0.493548	
1	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.624373	-0.624557	-0.657825	-0.630373	-0.625599	-0.535904	
2	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.598304	-0.599374	-0.631111	-0.604437	-0.599954	-0.402786	
3	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.572236	-0.574826	-0.594379	-0.565532	-0.569074	-0.332067	
4	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.563546	-0.565366	-0.584361	-0.552564	-0.548548	-0.334714	
...
144311	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.546167	-0.548369	-0.540951	-0.500692	-0.516038	-0.322360	
144312	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.520099	-0.522798	-0.497541	-0.448819	-0.473937	-0.315208	
144313	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.494030	-0.487858	-0.464149	-0.409914	-0.446880	-0.308242	
144314	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.467962	-0.466100	-0.434095	-0.377494	-0.413261	-0.295816	
144315	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.473755	-0.476641	0.013363	-0.387220	-0.466707	1.376792	

144316 rows × 18 columns

Start coding or [generate](#) with AI.

data.shape

(144316, 25)

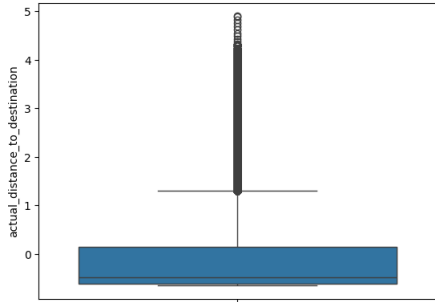
data.describe()

	data	route_type	source_center	source_name	destination_center	destination_name	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	actual_time	osrm_time	osrm_distance		
count	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000	144316.000000
mean	0.725020	0.686909	603.952812	621.334807	630.422767	612.400212	963.697698	233.561345	234.708498	417.996237	214.437055	285.549785	2.	
std	0.446506	0.463753	451.735582	413.862990	436.857113	414.612910	1038.082976	345.245823	345.480571	598.940065	308.448543	421.717826	1.	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	20.000000	9.000000	9.000045	9.000000	6.000000	9.008200	0.	
25%	0.000000	0.000000	135.000000	212.000000	197.000000	209.000000	161.000000	22.000000	23.352027	51.000000	27.000000	29.896250	1.	
50%	1.000000	1.000000	600.000000	538.000000	632.000000	520.000000	451.000000	66.000000	66.135322	132.000000	64.000000	78.624400	1.	
75%	1.000000	1.000000	913.000000	939.000000	901.000000	932.000000	1645.000000	286.000000	286.919294	516.000000	259.000000	346.305400	2.	
max	1.000000	1.000000	1495.000000	1495.000000	1465.000000	1465.000000	7898.000000	1927.000000	1927.447705	4532.000000	1686.000000	2326.199100	77.	

```
import seaborn as sns
```

```
sns.boxplot(y=trans['actual_distance_to_destination'])
```

<Axes: ylabel='actual_distance_to_destination'>



```
iqr=trans['actual_distance_to_destination'].quantile(0.75)-trans['actual_distance_to_destination'].quantile(0.25)
upper_limit=trans['actual_distance_to_destination'].quantile(0.75)+iqr
```

upper_limit

0.914028696880713

```
lower_limit=trans['actual_distance_to_destination'].quantile(0.25)-iqr
```

lower_limit

-1.3746804391333942

```
outliers =trans[(trans['actual_distance_to_destination']>upper_limit)][ 'actual_distance_to_destination']
```

outliers

```
384 0.976451
385 1.041219
386 1.103707
387 1.168056
388 1.231219
...
144245 3.984212
144246 4.048609
144247 4.097187
144248 4.164075
144249 4.211340
Name: actual_distance_to_destination, Length: 22550, dtype: float64
```

outliers

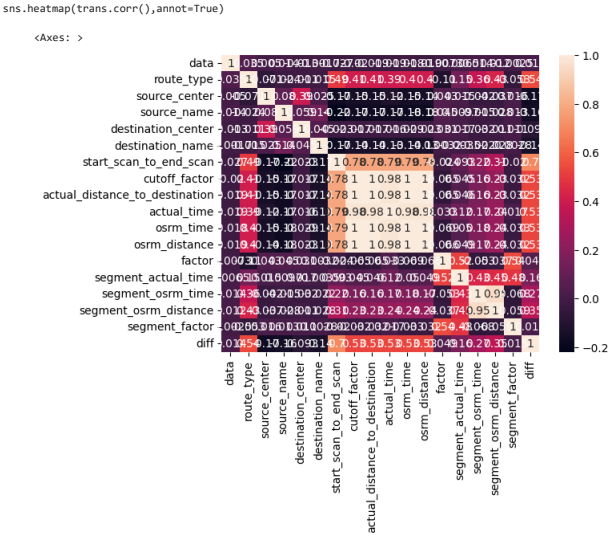
```
384      0.976451
385      1.041219
386      1.103707
387      1.168056
388      1.231219
...
144245  3.984212
144246  4.048609
144247  4.097187
144248  4.164875
144249  4.211348
Name: actual_distance_to_destination, Length: 22550, dtype: float64
```

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

array([1, 0])
```

```
s=trans.corr()
s
```

	data	route_type	source_center	source_name	destination_center	destination_name	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	actual_time	osrm_time	osrm_distance
data	1.000000	0.034511	0.005021	0.013609	-0.013263	-0.001667	-0.027282	-0.019694	-0.019211	-0.019282	-0.017857	-0.018959
route_type	0.034511	1.000000	-0.070741	-0.023693	-0.011122	-0.014905	0.491207	0.406246	0.406655	0.391543	0.402713	0.402213
source_center	0.005021	-0.070741	1.000000	0.079619	0.390113	0.025333	-0.168160	-0.148569	-0.148588	-0.120811	-0.148147	-0.144871
source_name	0.013609	-0.023693	0.079619	1.000000	0.058672	0.135591	-0.218436	-0.170355	-0.170501	-0.169362	-0.176565	-0.176547
destination_center	-0.013263	-0.011122	0.390113	0.058672	1.000000	0.044861	-0.023022	-0.016621	-0.016702	-0.016352	-0.029042	-0.023119
destination_name	-0.001667	-0.014905	0.025333	0.135591	0.044861	1.000000	-0.173401	-0.135854	-0.135979	-0.132676	-0.140176	-0.134886
start_scan_to_end_scan	-0.027282	0.491207	-0.168160	-0.218436	-0.023022	-0.173401	1.000000	0.784562	0.784895	0.785788	0.785189	0.784025
cutoff_factor	-0.019694	0.406246	-0.148569	-0.170355	-0.016621	-0.135854	0.784562	1.000000	0.999986	0.978745	0.995832	0.997114
actual_distance_to_destination	-0.019211	0.406655	-0.148588	-0.170501	-0.016702	-0.135979	0.784895	0.999986	1.000000	0.978683	0.995871	0.997114
actual_time	-0.019282	0.391543	-0.120811	-0.169362	-0.016352	-0.132676	0.785788	0.978745	0.978683	1.000000	0.978022	0.979425
osrm_time	-0.017857	0.402713	-0.148147	-0.176565	-0.029042	-0.140176	0.785189	0.995832	0.995871	0.978022	1.000000	0.999119
osrm_distance	-0.018959	0.402213	-0.144871	-0.176547	-0.023119	-0.134886	0.784025	0.997114	0.997147	0.979425	0.999119	1.000000
factor	-0.007305	-0.113972	0.043082	0.045267	0.031052	0.003214	-0.023648	-0.064684	-0.064869	0.033150	-0.069198	-0.065510
segment_actual_time	0.006469	0.147170	0.015206	0.009697	0.016779	-0.003488	0.083397	0.045333	0.045588	0.124549	0.050231	0.049055
segment_osrm_time	0.014099	0.363082	-0.041902	-0.015146	-0.032418	-0.022312	0.220957	0.158723	0.159614	0.172267	0.177813	0.169924
segment_osrm_distance	0.011762	0.428425	-0.036524	-0.028479	-0.011063	-0.028075	0.308057	0.231825	0.232832	0.243035	0.242969	0.240373
segment_factor	-0.002471	-0.053430	0.015762	0.013152	0.010766	0.002848	-0.020385	-0.031538	-0.031688	0.017420	-0.033136	-0.031885
diff	0.014306	0.541202	-0.168485	-0.159192	-0.092519	-0.141698	0.702066	0.528575	0.529021	0.530555	0.526719	0.525347



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

<ipython-Input-125-22e118d719e9>:1: 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
data['diff']=pd.to_datetime(data['od_end_time'])-pd.to_datetime(data['od_start_time'])
```

```
data.head(4)
data['trip_uid'].unique()

array(['trip-153741893647649320', 'trip-153768492602129387',
      'trip-153693976643699843', ..., 'trip-153761584139918815',
      'trip-153718412883843340', 'trip-153746866843555182'], dtype=object)
```

```
p=trans[['actual_time','osrm_time']]

p # created separate dataset in whic task related values are there
```

	actual_time	osrm_time	
0	-0.674521	-0.659552	
1	-0.657825	-0.630373	
2	-0.631111	-0.604437	
3	-0.594379	-0.565532	
4	-0.584361	-0.552564	
...	
144311	-0.540951	-0.500692	
144312	-0.497541	-0.448819	
144313	-0.464149	-0.409914	
144314	-0.434095	-0.377494	
144315	0.013363	-0.387220	

144316 rows × 2 columns

```
# based on z test we are going to find out
# ho: null hypothesis is osrm_time it is impractical assumption it will be far from original
# ha: alternate hypothesis is true value it will be opposite to null hypothesis

l=p['osrm_time'].mean() # l is null hypothesis mean value

l
-4.647803033326413e-17

o=p['actual_time'].mean() # o is alternate hypothesis mean value of it
o
-9.847040324844894e-18

# we are going to find alternate hypothesis z score value but std we have to find out
std=p['actual_time'].std()
std

1.0000034646373133

s=o-l
z_value=s/std

z_value
3.6630863095764934e-17

from scipy.stats import norm
p_value=1-norm.cdf(z_value)
p_value

0.5

# if p_value is greater than significant_level then null will be accepted that both actual_time and osrm time are same
# i am considering significant level as 5/100 =0.05

significant_level=10/100
if p_value>significant_level:
    print("null hypothesis is accepted")
else:
    print("alternate hypothesis will be accepted")

null hypothesis is accepted

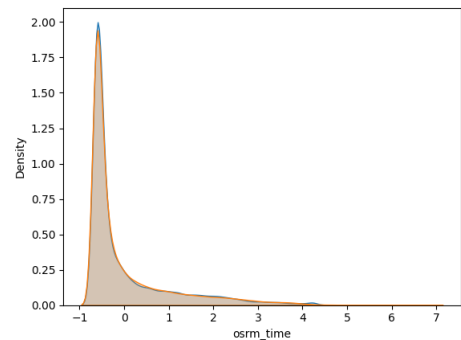
# hence osrm time and actual_time are approximately equal there is no significant difference

import seaborn as sns # for each column the distribution of data will be there here
sns.kdeplot(data=p['osrm_time'],shade=True,label='h0')
sns.kdeplot(data=p['actual_time'],shade=True,label='h1')

<ipython-input-91-f13c39cdf1cb>:2: FutureWarning:
`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

sns.kdeplot(data=p['osrm_time'],shade=True,label='h0')
<ipython-input-91-f13c39cdf1cb>:3: FutureWarning:
`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

sns.kdeplot(data=p['actual_time'],shade=True,label='h1')
<Axes: xlabel='osrm_time', ylabel='Density'>
```



```
# comparing difference between start_scan to end_scan by using z test
```

trans

	data	route_type	source_center	source_name	destination_center	destination_name	start_scan_to_end_scan	cutoff_factor	actual_distance_to_destination	actual_time	osrm_time	osrm_distance	factor	segment_ac
0	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.650441	-0.649164	-0.674521	-0.659552	-0.648741	-0.493548	
1	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.624373	-0.624557	-0.657825	-0.630373	-0.625599	-0.535904	
2	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.598304	-0.599374	-0.631111	-0.604437	-0.599954	-0.402786	
3	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.572236	-0.574826	-0.594379	-0.565532	-0.569074	-0.332067	
4	0.615851	-1.481203	-0.247829	-1.385330	-0.330596	0.305346	-0.845502	-0.563546	-0.565366	-0.584361	-0.552564	-0.548548	-0.334714	
...
144311	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.546167	-0.548369	-0.540951	-0.500692	-0.516038	-0.322360	
144312	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.520099	-0.522798	-0.497541	-0.448819	-0.473937	-0.315208	
144313	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.494030	-0.487858	-0.464149	-0.409914	-0.446880	-0.308242	
144314	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.467962	-0.466100	-0.434095	-0.377494	-0.413261	-0.295816	
144315	0.615851	-1.481203	-1.168725	1.763066	-1.429357	-0.222860	-0.517010	-0.473755	-0.476641	0.013363	-0.387220	-0.466707	1.376792	

144316 rows × 18 columns

actual time aggregated value and osrm time aggregated value

```
0      -0.845582
1      -0.845582
2      -0.845582
3      -0.845582
4      -0.845582
...
144311 -0.517818
144312 -0.517818
144313 -0.517818
144314 -0.517818
144315 -0.517818
Name: start_scan_to_end_scan, Length: 144316, dtype: float64
```

```
w=trans[['actual_time','segment_actual_time']]
w
```

	actual_time	segment_actual_time
0	-0.674521	-0.414306
1	-0.657825	-0.489039
2	-0.631111	-0.376940
3	-0.594379	-0.283524
4	-0.584361	-0.563772
...
144311	-0.540951	-0.451673
144312	-0.497541	-0.190108
144313	-0.464149	-0.302207
144314	-0.434095	-0.358257
144315	0.013363	4.331218

144316 rows × 2 columns

 Generate

Using ...

print hello world using rot13




Close

Generate is available for a limited time for unsubscribed users. [Upgrade to Colab Pro](#)




```
# null hypothesis is segment_actual_time
# actual time is alternate hypothesis
```

 Generate

Using ...

create a dataframe with 2 columns and 10 rows



Close

Generate is available for a limited time for unsubscribed users. [Upgrade to Colab Pro](#)



```
h=w['actual_time'].mean()
h

-9.847040324844894e-18

std1=w['actual_time'].std()

f=w['segment_actual_time'].mean()

z1=h-f
```