```
import pandas as pd

df=pd.read_csv('/content/drive/MyDrive/train.csv')

df.head()
```

	ID	Delivery_person_ID	Delivery_person_Age	Delivery_person_Ratings	Restaurant_latitude	Restaurant_longitude	Delivery_locatio
0	0x4607	INDORES13DEL02	37	4.9	22.745049	75.892471	
1	0xb379	BANGRES18DEL02	34	4.5	12.913041	77.683237	
2	0x5d6d	BANGRES19DEL01	23	4.4	12.914264	77.678400	
3	0x7a6a	COIMBRES13DEL02	38	4.7	11.003669	76.976494	
4	0x70a2	CHENRES12DEL01	32	4.6	12.972793	80.249982	

5 rows × 24 columns

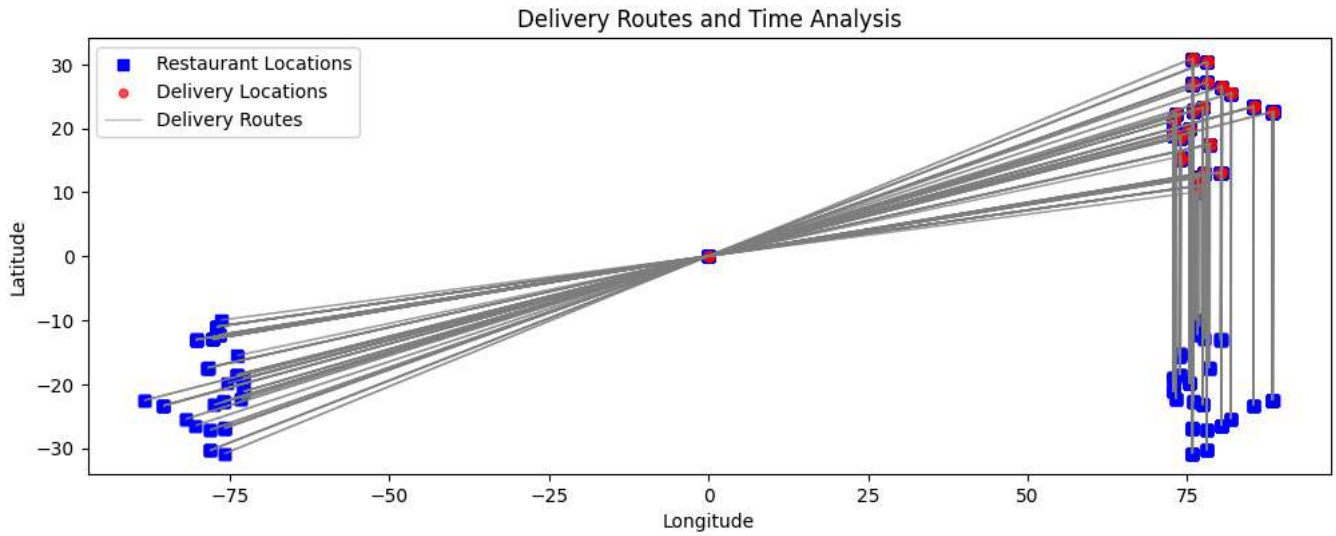
```
df['Delivery_location_latitude'].isnull().sum()
```

```
0
```

```
df['Delivery_location_longitude'].isnull().sum()
```

```
0
```

```
import geopandas as gpd
import matplotlib.pyplot as plt
from shapely.geometry import LineString


df_clean = df.dropna(subset=["Delivery_location_latitude", "Delivery_location_longitude",
                             "Restaurant_latitude", "Restaurant_longitude"])

# Create a GeoDataFrame for restaurant locations
gdf_rest = gpd.GeoDataFrame(df_clean, geometry=gpd.points_from_xy(df_clean["Restaurant_longitude"], df_clean["Restaurant_latitude"]))

# Create a GeoDataFrame for delivery locations
gdf_deliv = gpd.GeoDataFrame(df_clean, geometry=gpd.points_from_xy(df_clean["Delivery_location_longitude"], df_clean["Delivery_location_latit

# Create a GeoDataFrame for delivery routes (Lines connecting restaurant to delivery location)
df_clean["route"] = df_clean.apply(lambda x: LineString([(x["Restaurant_longitude"], x["Restaurant_latitude"]),
                                                         (x["Delivery_location_longitude"], x["Delivery_location_latitude"])]), axis=1)
gdf_routes = gpd.GeoDataFrame(df_clean, geometry="route")

# Plot the map
fig, ax = plt.subplots(figsize=(12, 8))

# Plot restaurant locations
gdf_rest.plot(ax=ax, marker="s", color="blue", markersize=30, label="Restaurant Locations")

# Plot delivery locations
gdf_deliv.plot(ax=ax, marker="o", color="red", markersize=20, alpha=0.7, label="Delivery Locations")

# Plot routes as lines
gdf_routes.plot(ax=ax, color="gray", linewidth=1, alpha=0.5, label="Delivery Routes")

# Add legend and title
plt.legend()
plt.title("Delivery Routes and Time Analysis")
plt.xlabel("Longitude")
plt.ylabel("Latitude")

# Show plot
plt.show()
```




```
import folium
from folium.plugins import HeatMap

# Create a base map centered at the average location
m = folium.Map(location=[df["Delivery_location_latitude"].mean(), df["Delivery_location_longitude"].mean()], zoom_start=12)

# Prepare heatmap data
heat_data = list(zip(df["Delivery_location_latitude"], df["Delivery_location_longitude"]))

# Add heatmap layer
HeatMap(heat_data).add_to(m)

# Save map as an interactive HTML file
m.save("delivery_demand_heatmap.html")
```

Start coding or generate with AI.