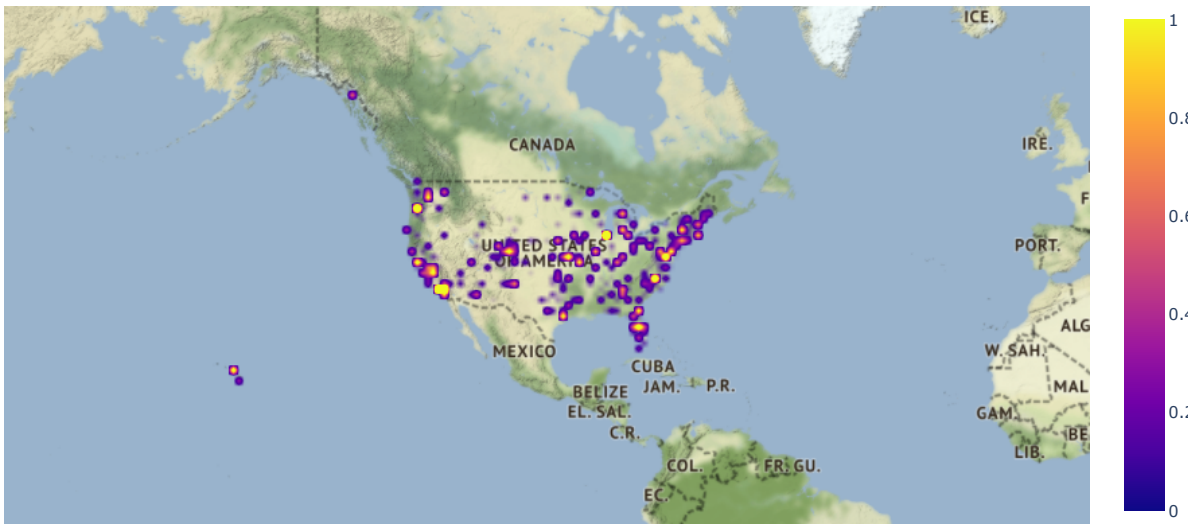
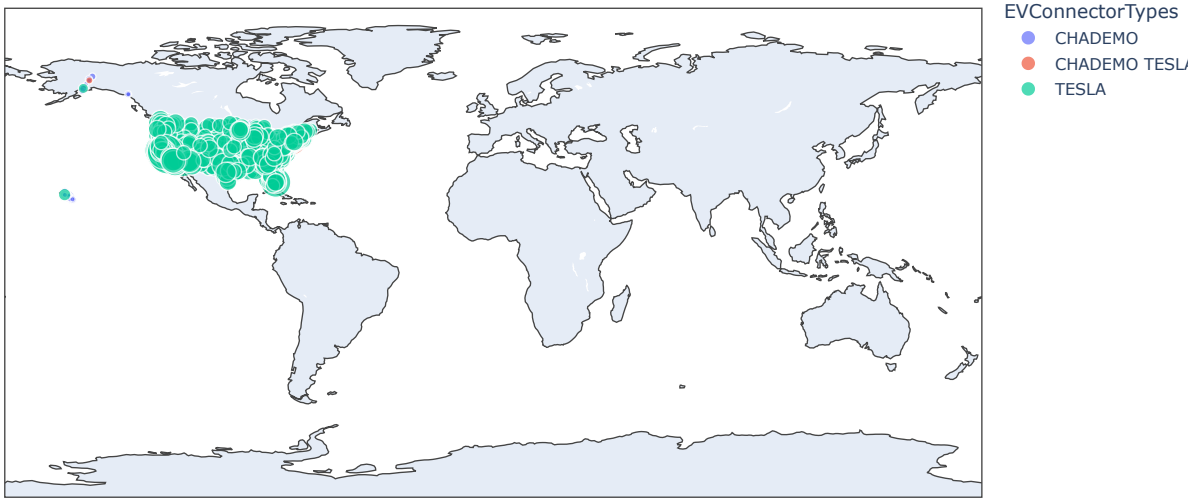
In [1]:
```
import pandas as pd
import plotly.express as px

#read in data for fast chargers
chargingstation_dc = pd.read_csv('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/alt_fuel_stations_dc.csv')

#create density map for fast chargers
fig = px.density_mapbox(chargingstation_dc,lat='Latitude',lon='Longitude',radius=1, zoom=1.5, mapbox_style = "stamen-terrain")
fig.show()
```



In [2]:
```
#create geography scatter map for fast chargers
fig = px.scatter_geo(chargingstation_dc,lat='Latitude',lon='Longitude',hover_name="City",color="EVConnectorTypes",size="EV.DC.Fast.Count")
fig.show()
```



In [3]:
```
chargingstation_dc = chargingstation_dc[chargingstation_dc.EVConnectorTypes != "TESLA"]
chargingstation_dc
```

Out[3]:

	Unnamed: 0	Fuel.Type.Code	Station.Name	Street.Address	Intersection.Directions	City	State	ZIP	Plus4	Station.Phone	...	EV.Pricing..French
0	1	ELEC	City of Greenville - Richardson Garage	66 Richardson St	NaN	Greenville	SC	29601	NaN	864-467-4900	...	NaN
1	2	ELEC	City of Santa Monica - Civic Parking Garage	333 Civic Center Dr	1st floor of parking garage	Santa Monica	CA	90401	NaN	310-458-8516	...	NaN
2	3	ELEC	Denver International Airport - Canopy Airport ...	8100 Tower Rd	Located in the indoor valet and covered self-p...	Commerce City	CO	80022	NaN	303-574-9800	...	NaN
3	4	ELEC	Clay Cooley Nissan	4914 S IH-35	NaN	Austin	TX	78745	NaN	512-444-1400	...	NaN
4	5	ELEC	Town North Nissan	9160A Research Blvd	NaN	Austin	TX	78758	NaN	512-451-7411	...	NaN
...	...	...	...	...	...	...	...	...	...	...	...	..
5997	5998	ELEC	Pepco - Forestville	8400 Old Malboro Pike	NaN	Upper Marlboro	MD	20770	NaN	855-900-7584	...	NaN
5998	5999	ELEC	Mountainside Fitness	1253 N Greenfield Rd	NaN	Mesa	AZ	85205	NaN	888-264-2208	...	NaN
5999	6000	ELEC	Wegman's	3195 Monroe Ave.	NaN	Rochester	NY	14618	NaN	888-264-2208	...	NaN
6000	6001	ELEC	Boise Fairview	8200 W Fairview Ave	NaN	Boise	ID	83704	NaN	888-264-2208	...	NaN
6001	6002	ELEC	GEORGIA POWER BREMEN DC	125 US-27	NaN	Bremen	GA	30110	NaN	888-758-4389	...	NaN

4476 rows × 66 columns

In [4]:
```
#read in data for level 2 and dc chargers, tesla filtered
chargingstation__lv2dc_filtered = pd.read_csv('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/alt_fuel_stations_lv2dc.filtered.csv')

#create denisty map for level 2 and dc chargers, tesla filtered
fig = px.density_mapbox(chargingstation__lv2dc_filtered,lat='Latitude',lon='Longitude',radius=1, zoom=1.5, mapbox_style = "stamen-terrain")
fig.show()
```

In [5]:
```
#create geography map of level 2 and fast chargers, tesla filter with size = total number of both types
fig = px.scatter_geo(chargingstation__lv2dc_filtered,lat='Latitude',lon='Longitude',hover_name="City",color="EV.Connector.Types",size="Level2.DC.Total")
fig.show()
```

In [6]:
```
#filter charging stations for Minnesota to account for travel locations on the border
chargingstation_dc_mn = chargingstation_dc[chargingstation_dc.State == "MN"]
chargingstation__lv2dc_filtered_mn = chargingstation__lv2dc_filtered[chargingstation__lv2dc_filtered.State == "MN"]
```

In [7]:
```
#filter charging stations for Iowa
chargingstation_dc_ia = chargingstation_dc[chargingstation_dc.State == "IA"]
chargingstation__lv2dc_filtered_ia = chargingstation__lv2dc_filtered[chargingstation__lv2dc_filtered.State == "IA"]
```

In [8]:
```
#filter charging stations for Illinois
chargingstation_dc_il = chargingstation_dc[chargingstation_dc.State == "IL"]
chargingstation__lv2dc_filtered_il = chargingstation__lv2dc_filtered[chargingstation__lv2dc_filtered.State == "IL"]
```

In [9]:
```
#filter charging stations for Michigan
chargingstation_dc_mi = chargingstation_dc[chargingstation_dc.State == "MI"]
chargingstation__lv2dc_filtered_mi = chargingstation__lv2dc_filtered[chargingstation__lv2dc_filtered.State == "MI"]
```

In [10]:
```
#filter charging stations for Wisconsin.
chargingstation_dc = chargingstation_dc[chargingstation_dc.State == "WI"]
chargingstation__lv2dc_filtered = chargingstation__lv2dc_filtered[chargingstation__lv2dc_filtered.State == "WI"]
```

In [11]:
```
#combine fast charging stations for all five states
chargingstation_dc = chargingstation_dc.append(chargingstation_dc_mn)
chargingstation_dc = chargingstation_dc.append(chargingstation_dc_ia)
chargingstation_dc = chargingstation_dc.append(chargingstation_dc_il)
chargingstation_dc = chargingstation_dc.append(chargingstation_dc_mi)
chargingstation_dc
```

Out[11]:

	Unnamed: 0	Fuel.Type.Code	Station.Name	Street.Address	Intersection.Directions	City	State	ZIP	Plus4	Station.Phone	...	EV.Pricing..French
328	329	ELEC	Rosen Nissan	2510 W Beltline Hwy	NaN	Madison	WI	53713	NaN	608-271-6000	...	NaN
329	330	ELEC	Zimbrick Nissan	5330 High Crossing Blvd	NaN	Madison	WI	53718	NaN	608-241-1122	...	NaN
470	471	ELEC	GMR Marketing	5000 S Towne Dr	NaN	New Berlin	WI	53151	NaN	262-786-5600	...	NaN
512	513	ELEC	MGE KELLEY E DCFC	3859 US-151	NaN	Madison	WI	53704	NaN	888-758-4389	...	NaN
1067	1068	ELEC	Ground Round Grill & Bar	201 Helen Walton St	NaN	Tomah	WI	54660	NaN	608-372-4000	...	NaN
...	...	...	...	...	...	...	...	...	...	...	...	..
5693	5694	ELEC	Meijer Royal Oak, MI #34	5111 Meijer Dr.	NaN	Royal Oak	MI	48073	NaN	877-455-3833	...	NaN
5917	5918	ELEC	West Michigan International	5380 International Dr	NaN	Kalamazoo	MI	49009	NaN	269-345-2183	...	NaN
5933	5934	ELEC	SQM STORE 206 SQM STORE 206 B	2020 Water St	NaN	Port Huron	MI	48060	NaN	888-758-4389	...	NaN
5934	5935	ELEC	SQM STORE 206 SQM STORE 206	2020 Water St	NaN	Port Huron	MI	48060	NaN	888-758-4389	...	NaN
5986	5987	ELEC	89 - Lansing DC Site	3309 South Creyts Road	NaN	Lansing	MI	48917	NaN	855-900-7584	...	NaN

349 rows × 66 columns

In [12]:
```
#combine level 2 & fast charging stations for all five states
chargingstation__lv2dc_filtered
chargingstation__lv2dc_filtered = chargingstation__lv2dc_filtered.append(chargingstation__lv2dc_filtered_mn)
chargingstation__lv2dc_filtered = chargingstation__lv2dc_filtered.append(chargingstation__lv2dc_filtered_ia)
chargingstation__lv2dc_filtered = chargingstation__lv2dc_filtered.append(chargingstation__lv2dc_filtered_il)
chargingstation__lv2dc_filtered = chargingstation__lv2dc_filtered.append(chargingstation__lv2dc_filtered_mi)
```

In [13]:
```
import json

#read file with hotels in Wisconsin
with open('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/osm_hotel.json', encoding="utf8") as f:
    hotelfile = json.load(f)
hotel = hotelfile['elements']

#read file with restaurants in Wisconsin
with open('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/osm_restaurant.json', encoding="utf8") as f:
    restaurantfile = json.load(f)
restaurant = restaurantfile['elements']

tourism = hotel + restaurant
```

In [14]:
```
#create denisty map for hotels & restaurants
fig = px.density_mapbox(tourism,lat='lat',lon='lon',radius=1, zoom=1.5, mapbox_style = "stamen-terrain")
fig.show()
```

In [15]:
```
#create denisty map for hotels
fig = px.density_mapbox(hotel,lat='lat',lon='lon',radius=1, zoom=1.5, mapbox_style = "stamen-terrain")
fig.show()
```

In [16]:
```
#create denisty map for restaurant
fig = px.density_mapbox(restaurant,lat='lat',lon='lon',radius=1, zoom=1.5, mapbox_style = "stamen-terrain")
fig.show()
```

In [17]:
```
#convert tourism, restaurant and hotel to dataframe
tourism = pd.DataFrame(tourism)
hotel = pd.DataFrame(hotel)
restaurant = pd.DataFrame(restaurant)
```

In [18]:
```
#combine station name, city and state to make a unique name
chargingstation_dc["StationNameCityState"] = chargingstation_dc["Station.Name"] + " " + chargingstation_dc["Street.Address"]
+ " " + chargingstation_dc["City"] + " " + chargingstation_dc["State"]
chargingstation__lv2dc_filtered["StationNameCityState"] = chargingstation__lv2dc_filtered["Station.Name"] + " " + chargingsta
tion__lv2dc_filtered["Street.Address"]+ " " + chargingstation__lv2dc_filtered["City"] + " " + chargingstation__lv2dc_filtered
["State"]

#concatenating unique station name, lat and long to create a consolidated location as accepted by havesine function
chargingstation_dc_loc = list(zip(chargingstation_dc['StationNameCityState'],chargingstation_dc.Latitude, chargingstation_dc.
Longitude))
chargingstation__lv2dc_filtered_loc = list(zip(chargingstation__lv2dc_filtered['StationNameCityState'],chargingstation__lv2dc
_filtered.Latitude, chargingstation__lv2dc_filtered.Longitude))
tourism_location = list(zip(tourism.lat,tourism.lon))
hotel_location = list(zip(hotel.lat,hotel.lon))
restaurant_location = list(zip(restaurant.lat,restaurant.lon))

#remove nan values from lists
tourism_loc = [x for x in tourism_location if str(x) != '(nan, nan)']
hotel_loc = [x for x in hotel_location if str(x) != '(nan, nan)']
restaurant_loc = [x for x in restaurant_location if str(x) != '(nan, nan)']
```

In [19]:
```
from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="geoapiExercises")

# defining a  function to find zip code for each latitude and longitude
def findzipcode(data):

    data_zipcode = []

    for i in data:

        latitude = str(i[0])
        longitude = str(i[1])
        location = geolocator.reverse(latitude+","+longitude, timeout=None)
        if location != None:
            address = location.raw['address']
            zipcode = address.get('postcode')
            if zipcode != None:
                zipcode_list = (latitude,longitude,zipcode)
                data_zipcode.append(zipcode_list)

    return data_zipcode
```

In [20]:
```
#run this and the next 3 cells for first run. This takes around 14 hours and should be done differently to improve performanc
e as a next step.
#find zip code for each tourism data set.
#hotel_zipcode = findzipcode(hotel_loc)
```

In [21]:
```
#restaurant_zipcode = findzipcode(restaurant_loc)
```

In [22]:
```
#tourism_zipcode = findzipcode(tourism_loc)
```

In [23]:
```
#write to file for rerun of code as findzipcode takes a very long time to run
#hotel_zipcode.to_csv("C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/hotel.csv")
#restaurant_zipcode.to_csv("C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/restaurant.csv")
#tourism_zipcode.to_csv("C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/tourism.csv")
```

In [24]:
```
#read in files vs. running findzipcode - after findzipcode has been run the 1st time
hotel_zipcode = pd.read_csv('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/hotel.csv')
restaurant_zipcode = pd.read_csv('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/restaurant.csv')
tourism_zipcode = pd.read_csv('C:/Users/sbace/Documents/DataScience/DS785_Capstone/DataSources/Good/tourism.csv')
```

In [25]:
```
#group by zipcode and count
hotel_zipcode = pd.DataFrame(hotel_zipcode, columns = ['lat','lon','zipcode'])
hotel_count = hotel_zipcode.groupby('zipcode').count()

restaurant_zipcode = pd.DataFrame(restaurant_zipcode, columns = ['lat','lon','zipcode'])
restaurant_count = restaurant_zipcode.groupby('zipcode').count()

tourism_zipcode = pd.DataFrame(tourism_zipcode, columns = ['lat','lon','zipcode'])
tourism_count = tourism_zipcode.groupby('zipcode').count()
```

In [26]:
```
#filter by x number within zipcodes
hotel_zipcode_index = hotel_count[hotel_count > hotel_count.lat.median()].index.tolist()
restaurant_zipcode_index = restaurant_count[restaurant_count > restaurant_count.lat.median()].index.tolist()
tourism_zipcode_index = tourism_count[tourism_count > tourism_count.lat.median()].index.tolist()
```

In [27]:
```
hotel_count.lat.median(),restaurant_count.lat.median(),tourism_count.lat.median()
```

Out[27]: (26.0, 10.0, 15.0)

In [28]:
```
len(hotel_zipcode), len(restaurant_zipcode),len(tourism_zipcode)
```

Out[28]: (18913, 18264, 37177)

In [29]:
```
#function to average latitude & longitude to find one location per zipcode.
#As a next step, a more accurate central location would be to use gis centroid functionality from census data.

def findlatlon(index,data):

    data_zipcode = []

    for i in index:

        zipcode_filtered = data[data.zipcode == i]

        zipcode_filtered.lat = [float(x) for x in zipcode_filtered.lat]
        calclat = sum(zipcode_filtered.lat) / len(zipcode_filtered.lat)

        zipcode_filtered.lon = [float(x) for x in zipcode_filtered.lon]
        calclon = sum(zipcode_filtered.lon) / len(zipcode_filtered.lon)

        latlon = (i,calclat,calclon)
        data_zipcode.append(latlon)

    data_zipcode = pd.DataFrame(data_zipcode,columns=['zipcode','lat','lon'])
    data_zipcode['latlon'] = list(zip(data_zipcode.lat,data_zipcode.lon))

    return data_zipcode
```

In [30]:
```
#average latitude & longitude to find one location per zipcode
hotel_zipcode_latlon = findlatlon(hotel_zipcode_index,hotel_zipcode)
restaurant_zipcode_latlon = findlatlon(restaurant_zipcode_index,restaurant_zipcode)
tourism_zipcode_latlon = findlatlon(tourism_zipcode_index,tourism_zipcode)
```

```
C:\Users\sbace\Anaconda3\lib\site-packages\pandas\core\generic.py:5208: 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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [31]:
```
import haversine as hs
import folium

#defining a  function to calculate distance between two locations
#Loc1 hotel, restaurant or tourism location
#Loc2 charging station

def distance_from(loc1,loc2):
    dist=hs.haversine(loc1,loc2)
    return round(dist,2)
```

In [32]:
```
#convert charging station to data frame
chargingstation__lv2dc_filtered_loc = pd.DataFrame(chargingstation__lv2dc_filtered_loc,columns=['stationNameCityState','lat','lon'])
chargingstation__lv2dc_filtered_loc['latlon'] = list(zip(chargingstation__lv2dc_filtered_loc.lat,chargingstation__lv2dc_filtered_loc.lon))

chargingstation_dc_loc = pd.DataFrame(chargingstation_dc_loc,columns=['stationNameCityState','lat','lon'])
chargingstation_dc_loc['latlon'] = list(zip(chargingstation_dc_loc.lat,chargingstation_dc_loc.lon))
```

In [33]:
```
#running a loop which will calculate distance from hotels to each level 2 and fast charging station
for _,row in chargingstation__lv2dc_filtered_loc.iterrows():
    hotel_zipcode_latlon[row.stationNameCityState] = hotel_zipcode_latlon['latlon'].apply(lambda x: distance_from(row.latlon,x))

#running a loop which will calculate distance from restaurants to each fast charging station
for _,row in chargingstation_dc_loc.iterrows():
    restaurant_zipcode_latlon[row.stationNameCityState] = restaurant_zipcode_latlon['latlon'].apply(lambda x: distance_from(row.latlon,x))

#running a loop which will calculate distance from hotels and restaurants to each fast charging station
for _,row in chargingstation_dc_loc.iterrows():
    tourism_zipcode_latlon[row.stationNameCityState] = tourism_zipcode_latlon['latlon'].apply(lambda x: distance_from(row.latlon,x))
```

In [34]:
```
#select min distance per row
col_four = hotel_zipcode_latlon.columns[4]
hotel_zipcode_latlon['dist_min'] = hotel_zipcode_latlon.loc[:,col_four:].min(axis=1)
hotel_zipcode_latlon['dist_to'] = hotel_zipcode_latlon.loc[:,col_four:].idxmin(axis=1)

col_four = restaurant_zipcode_latlon.columns[4]
restaurant_zipcode_latlon['dist_min'] = restaurant_zipcode_latlon.loc[:,col_four:].min(axis=1)
restaurant_zipcode_latlon['dist_to'] = restaurant_zipcode_latlon.loc[:,col_four:].idxmin(axis=1)

col_four = tourism_zipcode_latlon.columns[4]
tourism_zipcode_latlon['dist_min'] = tourism_zipcode_latlon.loc[:,col_four:].min(axis=1)
tourism_zipcode_latlon['dist_to'] = tourism_zipcode_latlon.loc[:,col_four:].idxmin(axis=1)
```

In [35]:
```
#find nearest charging station name, latitude and longitde for each hotel and/or restaurant
for i,row in hotel_zipcode_latlon.iterrows():

    findstation = chargingstation__lv2dc_filtered_loc[chargingstation__lv2dc_filtered_loc.stationNameCityState == row['dist_t
o']]

    hotel_zipcode_latlon.at[i,'dist_lat'] = findstation.lat.values[0]
    hotel_zipcode_latlon.at[i,'dist_lon'] = findstation.lon.values[0]

for i,row in restaurant_zipcode_latlon.iterrows():

    findstation = chargingstation_dc_loc[chargingstation_dc_loc.stationNameCityState == row['dist_to']]

    restaurant_zipcode_latlon.at[i,'dist_lat'] = findstation.lat.values[0]
    restaurant_zipcode_latlon.at[i,'dist_lon'] = findstation.lon.values[0]

for i,row in tourism_zipcode_latlon.iterrows():

    findstation = chargingstation_dc_loc[chargingstation_dc_loc.stationNameCityState == row['dist_to']]

    tourism_zipcode_latlon.at[i,'dist_lat'] = findstation.lat.values[0]
    tourism_zipcode_latlon.at[i,'dist_lon'] = findstation.lon.values[0]
```

In [36]:
```
hotel_zipcode_latlon
```

Out[36]:

	zipcode	lat	lon	latlon	Rosen Nissan 2510 W Beltline Hwy Madison WI	Zimbrick Nissan 5330 High Crossing Blvd Madison WI	GMR Marketing 5000 S Towne Dr New Berlin WI	MGE KELLEY E DCFC 3859 US-151 Madison WI	Ground Round Grill & Bar 201 Helen Walton St Tomah WI	Walmart Sam's Club 8185 4001 Gateway Dr Eau Claire WI	...	Ford Allen Park Charger 8500 Enterprise Dr Allen Park MI	West Michigan International 5380 International Dr Kalamazoo MI	SQM STORE 206 SQM STORE 206 B 2020 Water St Port Huron MI
0	49682	46.440025	-86.656762	(46.44002456538461, -86.65676243846156)	436.87	421.47	404.36	424.63	404.26	414.70	...	538.96	473.43	507.58
1	49801	45.810665	-88.035986	(45.81066544499999, -88.03598644500002)	327.50	313.03	317.72	315.94	278.59	289.26	...	551.35	439.21	543.34
2	49829	45.755277	-87.090365	(45.755277322448975, -87.09036537959182)	354.50	339.00	321.94	342.18	331.34	356.52	...	496.75	406.74	480.17
3	49855	46.544867	-87.417503	(46.544866811500015, -87.41750276600006)	420.91	406.18	403.06	409.16	370.64	368.58	...	581.33	498.08	556.72
4	49858	45.119227	-87.612741	(45.119226775, -87.61274114999999)	272.95	257.41	244.08	260.60	259.99	302.69	...	475.24	355.65	475.66
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
319	61080	42.489765	-89.001523	(42.489765025, -89.00152309375)	69.78	76.09	89.03	74.75	209.22	320.44	...	476.15	273.55	537.85
320	61085	42.358748	-90.005480	(42.3587485, -90.00547990000001)	89.20	104.86	168.27	101.62	188.97	292.07	...	558.58	355.20	621.50
321	61103	42.315001	-89.071341	(42.31500111612903, -89.07134072258066)	85.17	93.54	105.76	91.82	222.36	332.98	...	481.93	278.37	546.59
322	61111	42.323158	-88.969220	(42.323158275000004, -88.96921959166666)	87.50	94.60	99.09	93.14	226.05	337.00	...	473.51	269.99	538.17
323	61114	42.321559	-88.967454	(42.32155866, -88.96745408000001)	87.72	94.81	99.11	93.35	226.28	337.23	...	473.37	269.84	538.06

324 rows × 439 columns

In [37]:
```
#set variables for minimum, median, and maximum for published ranges and reductions in ranges due to medium and high impacts
stdmin = 170
stdmed = 245
stdmax = 361
medmin = 127.5
medmed = 183.8
medmax = 270.8
highmin = 85
highmed = 122.5
highmax = 180.5

#set boolean for standard ranges min/mean/max
hotel_zipcode_latlon['std_min'] = hotel_zipcode_latlon['dist_min'] >= stdmin
hotel_zipcode_latlon['std_med'] = hotel_zipcode_latlon['dist_min'] >= stdmed
hotel_zipcode_latlon['std_max'] = hotel_zipcode_latlon['dist_min'] >= stdmax

#set boolean for medium impact reduction to ranges min/mean/max
hotel_zipcode_latlon['med_min'] = hotel_zipcode_latlon['dist_min'] >= medmin
hotel_zipcode_latlon['med_med'] = hotel_zipcode_latlon['dist_min'] >= medmed
hotel_zipcode_latlon['med_max'] = hotel_zipcode_latlon['dist_min'] >= medmax

#set boolean for high impact reduction to ranges min/mean/max
hotel_zipcode_latlon['high_min'] = hotel_zipcode_latlon['dist_min'] >= highmin
hotel_zipcode_latlon['high_med'] = hotel_zipcode_latlon['dist_min'] >= highmed
hotel_zipcode_latlon['high_max'] = hotel_zipcode_latlon['dist_min'] >= highmax

#set boolean for standard ranges min/mean/max
restaurant_zipcode_latlon['std_min'] = restaurant_zipcode_latlon['dist_min'] >= stdmin
restaurant_zipcode_latlon['std_med'] = restaurant_zipcode_latlon['dist_min'] >= stdmed
restaurant_zipcode_latlon['std_max'] = restaurant_zipcode_latlon['dist_min'] >= stdmax

#set boolean for medium impact reduction to ranges min/mean/max
restaurant_zipcode_latlon['med_min'] = restaurant_zipcode_latlon['dist_min'] >= medmin
restaurant_zipcode_latlon['med_med'] = restaurant_zipcode_latlon['dist_min'] >= medmed
restaurant_zipcode_latlon['med_max'] = restaurant_zipcode_latlon['dist_min'] >= medmax

#set boolean for high impact reduction to ranges min/mean/max
restaurant_zipcode_latlon['high_min'] = restaurant_zipcode_latlon['dist_min'] >= highmin
restaurant_zipcode_latlon['high_med'] = restaurant_zipcode_latlon['dist_min'] >= highmed
restaurant_zipcode_latlon['high_max'] = restaurant_zipcode_latlon['dist_min'] >= highmax

#set boolean for standard ranges min/mean/max
tourism_zipcode_latlon['std_min'] = tourism_zipcode_latlon['dist_min'] >= stdmin
tourism_zipcode_latlon['std_med'] = tourism_zipcode_latlon['dist_min'] >= stdmed
tourism_zipcode_latlon['std_max'] = tourism_zipcode_latlon['dist_min'] >= stdmax

#set boolean for medium impact reduction to ranges min/mean/max
tourism_zipcode_latlon['med_min'] = tourism_zipcode_latlon['dist_min'] >= medmin
tourism_zipcode_latlon['med_med'] = tourism_zipcode_latlon['dist_min'] >= medmed
tourism_zipcode_latlon['med_max'] = tourism_zipcode_latlon['dist_min'] >= medmax

#set boolean for high impact reduction to ranges min/mean/max
tourism_zipcode_latlon['high_min'] = tourism_zipcode_latlon['dist_min'] >= highmin
tourism_zipcode_latlon['high_med'] = tourism_zipcode_latlon['dist_min'] >= highmed
tourism_zipcode_latlon['high_max'] = tourism_zipcode_latlon['dist_min'] >= highmax
```

In [38]:
```
hotel_zipcode_latlon
```

Out[38]:

	zipcode	lat	lon	latlon	Rosen Nissan 2510 W Beltline Hwy Madison WI	Zimbrick Nissan 5330 High Crossing Blvd Madison WI	GMR Marketing 5000 S Towne Dr New Berlin WI	MGE KELLEY E DCFC 3859 US-151 Madison WI	Ground Round Grill & Bar 201 Helen Walton St Tomah WI	Walmart Sam's Club 8185 4001 Gateway Dr Eau Claire WI	...	dist_lon	std_min	std_med	st
0	49682	46.440025	-86.656762	(46.44002456538461, -86.65676243846156)	436.87	421.47	404.36	424.63	404.26	414.70	...	-87.086401	False	False	
1	49801	45.810665	-88.035986	(45.81066544499999, -88.03598644500002)	327.50	313.03	317.72	315.94	278.59	289.26	...	-87.907667	False	False	
2	49829	45.755277	-87.090365	(45.755277322448975, -87.09036537959182)	354.50	339.00	321.94	342.18	331.34	356.52	...	-87.086401	False	False	
3	49855	46.544867	-87.417503	(46.544866811500015, -87.41750276600006)	420.91	406.18	403.06	409.16	370.64	368.58	...	-87.086401	False	False	
4	49858	45.119227	-87.612741	(45.119226775, -87.61274114999999)	272.95	257.41	244.08	260.60	259.99	302.69	...	-87.996011	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
319	61080	42.489765	-89.001523	(42.489765025, -89.00152309375)	69.78	76.09	89.03	74.75	209.22	320.44	...	-88.978290	False	False	
320	61085	42.358748	-90.005480	(42.3587485, -90.00547990000001)	89.20	104.86	168.27	101.62	188.97	292.07	...	-90.714196	False	False	
321	61103	42.315001	-89.071341	(42.31500111612903, -89.07134072258066)	85.17	93.54	105.76	91.82	222.36	332.98	...	-88.978290	False	False	
322	61111	42.323158	-88.969220	(42.323158275000004, -88.96921959166666)	87.50	94.60	99.09	93.14	226.05	337.00	...	-88.978290	False	False	
323	61114	42.321559	-88.967454	(42.32155866, -88.96745408000001)	87.72	94.81	99.11	93.35	226.28	337.23	...	-88.978290	False	False	

324 rows × 448 columns

In [39]:
```
restaurant_zipcode_latlon
```

Out[39]:

	zipcode	lat	lon	latlon	Rosen Nissan 2510 W Beltline Hwy Madison WI	Zimbrick Nissan 5330 High Crossing Blvd Madison WI	GMR Marketing 5000 S Towne Dr New Berlin WI	MGE KELLEY E DCFC 3859 US-151 Madison WI	Ground Round Grill & Bar 201 Helen Walton St Tomah WI	Walmart Sam's Club 8185 4001 Gateway Dr Eau Claire WI	...	dist_lon	std_min	std_med	st
0	10175	42.966364	-88.048639	(42.96636447777777, -88.04863920000001)	111.89	102.57	5.46	104.58	230.31	337.35	...	-88.113495	False	False	
1	49801	45.830216	-88.059208	(45.83021606, -88.05920846000001)	328.93	314.51	319.86	317.41	278.86	288.43	...	-87.907667	False	False	
2	49802	45.798922	-88.071883	(45.7989217, -88.0718831)	325.32	310.90	316.37	313.80	275.70	286.19	...	-87.907667	False	False	
3	49808	46.815598	-87.728702	(46.81559755, -87.72870244999999)	440.82	426.59	430.47	429.44	379.19	365.67	...	-87.907667	False	False	
4	49815	46.199283	-88.019384	(46.1992834, -88.0193844)	368.77	354.55	360.95	357.40	311.37	309.33	...	-87.907667	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
638	61111	42.318341	-89.041879	(42.31834081639344, -89.04187919508196)	85.66	93.66	103.74	92.02	223.31	334.03	...	-88.978290	False	False	
639	61114	42.307879	-88.984021	(42.30787947326732, -88.98402092673267)	88.55	95.91	101.15	94.40	226.82	337.69	...	-88.978290	False	False	
640	61115	42.353387	-89.044206	(42.353386633333336, -89.04420633333335)	81.96	89.81	101.25	88.20	219.90	330.71	...	-88.978290	False	False	
641	63719	43.016237	-89.510471	(43.01623659999999, -89.5104715)	7.59	22.77	113.84	19.46	137.43	248.69	...	-89.421270	False	False	
642	92054	44.283164	-88.415319	(44.28316358571429, -88.41531925714285)	160.52	145.09	149.79	148.25	169.57	245.01	...	-89.407096	False	False	

643 rows × 366 columns

In [40]:
```
tourism_zipcode_latlon
```

Out[40]:

	zipcode	lat	lon	latlon	Rosen Nissan 2510 W Beltline Hwy Madison WI	Zimbrick Nissan 5330 High Crossing Blvd Madison WI	GMR Marketing 5000 S Towne Dr New Berlin WI	MGE KELLEY E DCFC 3859 US-151 Madison WI	Ground Round Grill & Bar 201 Helen Walton St Tomah WI	Walmart Sam's Club 8185 4001 Gateway Dr Eau Claire WI	...	dist_lon	std_min	std_med	st
0	10175	42.966364	-88.048639	(42.96636447777777, -88.04863920000001)	111.89	102.57	5.46	104.58	230.31	337.35	...	-88.113495	False	False	
1	49682	46.440025	-86.656762	(46.44002456538461, -86.65676243846156)	436.87	421.47	404.36	424.63	404.26	414.70	...	-87.086401	False	False	
2	49801	45.817182	-88.043727	(45.81718231666667, -88.04372711666666)	327.97	313.52	318.43	316.43	278.68	288.98	...	-87.907667	False	False	
3	49802	45.798922	-88.071883	(45.7989217, -88.0718831)	325.32	310.90	316.37	313.80	275.70	286.19	...	-87.907667	False	False	
4	49808	46.815598	-87.728702	(46.81559755, -87.72870244999999)	440.82	426.59	430.47	429.44	379.19	365.67	...	-87.907667	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
667	61111	42.319133	-89.029935	(42.31913272739726, -89.0299351506849)	85.93	93.79	102.96	92.18	223.75	334.52	...	-88.978290	False	False	
668	61114	42.309112	-88.982528	(42.30911183243245, -88.98252841801803)	88.47	95.81	100.97	94.31	226.77	337.65	...	-88.978290	False	False	
669	61115	42.353387	-89.044206	(42.353386633333336, -89.04420633333335)	81.96	89.81	101.25	88.20	219.90	330.71	...	-88.978290	False	False	
670	63719	43.016237	-89.510471	(43.01623659999999, -89.5104715)	7.59	22.77	113.84	19.46	137.43	248.69	...	-89.421270	False	False	
671	92054	44.283164	-88.415319	(44.28316358571429, -88.41531925714285)	160.52	145.09	149.79	148.25	169.57	245.01	...	-89.407096	False	False	

672 rows × 366 columns

In [41]:
```
hotel_std_min = hotel_zipcode_latlon[hotel_zipcode_latlon['std_min'] == True]
hotel_std_med = hotel_zipcode_latlon[hotel_zipcode_latlon['std_med'] == True]
hotel_std_max = hotel_zipcode_latlon[hotel_zipcode_latlon['std_max'] == True]
hotel_med_min = hotel_zipcode_latlon[hotel_zipcode_latlon['med_min'] == True]
hotel_med_med = hotel_zipcode_latlon[hotel_zipcode_latlon['med_med'] == True]
hotel_med_max = hotel_zipcode_latlon[hotel_zipcode_latlon['med_max'] == True]
hotel_high_min = hotel_zipcode_latlon[hotel_zipcode_latlon['high_min'] == True]
hotel_high_med = hotel_zipcode_latlon[hotel_zipcode_latlon['high_med'] == True]
hotel_high_max = hotel_zipcode_latlon[hotel_zipcode_latlon['high_max'] == True]

restaurant_std_min = restaurant_zipcode_latlon[restaurant_zipcode_latlon['std_min'] == True]
restaurant_std_med = restaurant_zipcode_latlon[restaurant_zipcode_latlon['std_med'] == True]
restaurant_std_max = restaurant_zipcode_latlon[restaurant_zipcode_latlon['std_max'] == True]
restaurant_med_min = restaurant_zipcode_latlon[restaurant_zipcode_latlon['med_min'] == True]
restaurant_med_med = restaurant_zipcode_latlon[restaurant_zipcode_latlon['med_med'] == True]
restaurant_med_max = restaurant_zipcode_latlon[restaurant_zipcode_latlon['med_max'] == True]
restaurant_high_min = restaurant_zipcode_latlon[restaurant_zipcode_latlon['high_min'] == True]
restaurant_high_med = restaurant_zipcode_latlon[restaurant_zipcode_latlon['high_med'] == True]
restaurant_high_max = restaurant_zipcode_latlon[restaurant_zipcode_latlon['high_max'] == True]

tourism_std_min = tourism_zipcode_latlon[tourism_zipcode_latlon['std_min'] == True]
tourism_std_med = tourism_zipcode_latlon[tourism_zipcode_latlon['std_med'] == True]
tourism_std_max = tourism_zipcode_latlon[tourism_zipcode_latlon['std_max'] == True]
tourism_med_min = tourism_zipcode_latlon[tourism_zipcode_latlon['med_min'] == True]
tourism_med_med = tourism_zipcode_latlon[tourism_zipcode_latlon['med_med'] == True]
tourism_med_max = tourism_zipcode_latlon[tourism_zipcode_latlon['med_max'] == True]
tourism_high_min = tourism_zipcode_latlon[tourism_zipcode_latlon['high_min'] == True]
tourism_high_med = tourism_zipcode_latlon[tourism_zipcode_latlon['high_med'] == True]
tourism_high_max = tourism_zipcode_latlon[tourism_zipcode_latlon['high_max'] == True]
```

In [42]:
```
#map creation of charging station locations along with hotels and/or restaurants outside the range of the nearest charging st
ation

def create_map(poi,stations):
    m = folium.Map(location=[poi.lat.mean(), poi.lon.mean()], zoom_start=12, tiles='OpenStreetMap')

    for _, row in poi.iterrows():

        if row['std_max']==True:
            cluster_colour='brown'
            r = medmin
            f = True
        elif row['med_max']==True:
            cluster_colour = 'black'
            r = medmed
            f = True
        elif row['std_med']==True:
            cluster_colour = 'red'
            r = medmax
            f = True
        elif row['med_med']==True:
            cluster_colour = 'yellow'
            r = medmax
            f = True
        elif row['high_max']==True:
            cluster_colour = 'blue'
            r = medmed
            f = True
        elif row['std_min']==True:
            cluster_colour = 'green'
            r = medmax
            f = True
        elif row['med_min']==True:
            cluster_colour = 'purple'
            r = medmax
            f = True
        elif row['high_med']==True:
            cluster_colour = 'orange'
            r = medmed
            f = True
        elif row['high_min']==True:
            cluster_colour = 'maroon'
            r = medmax
            f = True
        else:
            cluster_colour = None
            r = 0
            f = False

        folium.CircleMarker(
            location= [row['lat'],row['lon']],
            radius=5,
            popup= row['zipcode'],
            color=cluster_colour,
            fill=f,
            fill_color=cluster_colour
            ).add_to(m)

    for _, row in stations.iterrows():
        folium.Marker(
            location= [row['lat'],row['lon']],
            #radius=1,
            popup= row['stationNameCityState'],
            color='blue',
            fill=True,
            fill_color='blue'
             ).add_to(m)
    for _, row in stations.iterrows():
        folium.Circle(
            location= [row['lat'],row['lon']],
            #radius=1,
            popup= row['stationNameCityState'],
            color='black',
            fill=False,
        ).add_to(m)

    return m
```

In [43]:
```
#create map of fast charging stations and restaurants outside the range of the nearest charging stations
create_map(restaurant_zipcode_latlon,chargingstation_dc_loc)
```

Out[43]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [44]:
```
#create map of fast charging stations and restaurants and hotels outside the range of the nearest charging stations
create_map(tourism_zipcode_latlon,chargingstation_dc_loc)
```

Out[44]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [45]:
```
#create map of level 2 and fast charging stations and hotels outside the range of the nearest charging stations
create_map(hotel_zipcode_latlon,chargingstation__lv2dc_filtered_loc)
```

Out[45]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [ ]:
```

```