

In [1]:

```
#python bibliotekų importavimas
import pandas as pd
import numpy as np
import arcgis
from arcgis.gis import GIS
gis = GIS()
from arcgis.geocoding import geocode
from shapely.geometry import Point
import folium
from datetime import datetime
from folium import plugins
from folium.plugins import HeatMap
```

C:\Users\Rokas\Anaconda3\lib\site-packages\arcgis\features_data\geodatabase\utils.py:16: FutureWarning: The pandas.datetime class is deprecated and will be removed from pandas in a future version. Import from datetime module instead.

```
pd.datetime,
```

In [2]:

```
# Lokacijų csv failo importavimas į python (#cp skirtas lietuviškai rasmenim skaityti)
df = r'C:\Porfolio\citybee_keliones\citybee_vietos.csv'
data = pd.read_csv(df, sep=';', encoding='cp775')
data.head()
```

Out[2]:

	id	title	address
0	2	Autosalonas "Krasta Auto Vilnius" - Vilnius	Ozo g. 10A, Vilnius, Lithuania
1	8	PC "Maxima", Ukmerges g. - Vilnius	Ukmerges g. 256, Vilnius, Lithuania
2	10	PC "Mada" - Vilnius	Virsuliskiu g. 42, 05112, Vilnius, Lithuania
3	12	"Ergo", Gelezinio Vilko g. - Vilnius	Gelezinio Vilko g. 6A, Vilnius, Lithuania
4	13	PC "Maxima", V. Grybo g. - Vilnius	Grybo g. 21, Vilnius, Lithuania

In [3]:

```
#Adreso geokodavimas ir x, y koordinatų ištraukimas
data['data'] = data.apply(lambda row: geocode(row['address']),axis=1)
data['x'] = data.apply(lambda row: (row['data'][0]['location']['x']),axis=1)
data['y'] = data.apply(lambda row: (row['data'][0]['location']['y']),axis=1)
data = data[['id','title','address','x','y']]
data.head()
```

Out[3]:

	id	title	address	x	y
0	2	Autosalonas "Krasta Auto Vilnius" - Vilnius	Ozo g. 10A, Vilnius, Lithuania	25.27953	54.71388
1	8	PC "Maxima", Ukmerges g. - Vilnius	Ukmerges g. 256, Vilnius, Lithuania	25.23978	54.72684
2	10	PC "Mada" - Vilnius	Virsuliskiu g. 42, 05112, Vilnius, Lithuania	25.22765	54.70762
3	12	"Ergo", Gelezinio Vilko g. - Vilnius	Gelezinio Vilko g. 6A, Vilnius, Lithuania	25.23741	54.67211
4	13	PC "Maxima", V. Grybo g. - Vilnius	Grybo g. 21, Vilnius, Lithuania	25.31451	54.70032

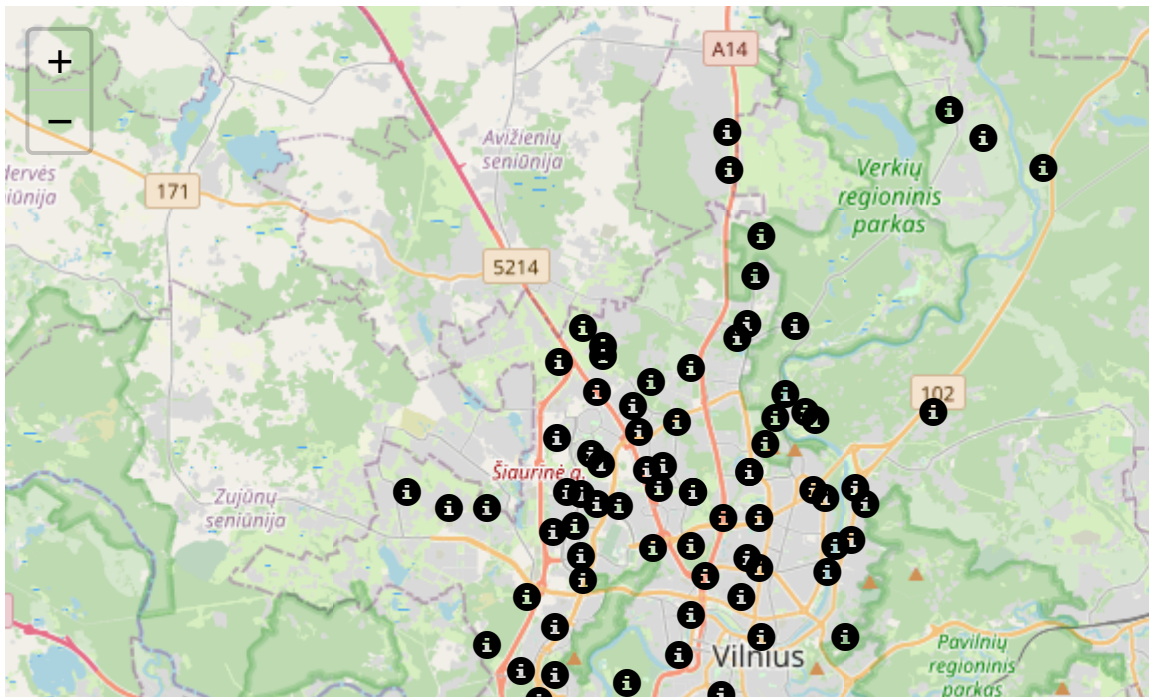
In [4]:

```
#Lokacijų atvaizdavimas žemėlapyje naudojant Folium biblioteką
m = folium.Map(location=[54.71378495, 25.2794343068304],
                  width=750, height=500,
                  zoom_start = 12,
                  control_scale=True)

for g in data.itertuples():
    folium.Marker(location=[g.y, g.x],
                  popup= folium.Popup(g.title, max_width=500, min_width=200),
                  icon = folium.Icon(color = 'green',size = (10,10))).add_to(m)

m
```

Out[4]:



In [5]:

```
# Kelionių csv failo importavimas į python (#cp skirtas lietuviškom rasmenim skaityti)
df2 = r'C:\Porfolio\citybee_keliones\citybee_duomenys.csv'
kelioniu_data = pd.read_csv(df2, sep=';', encoding='cp775')
kelioniu_data.head()
```

Out[5]:

	id	customer_id	plate	start	end	duration_min	distance_km	createdat	zone_
0	1312776	27617	JNG718	2017-07-11 19:01	NaN	0	0	2017-07-11 19:01	
1	1312332	47748	JFD283	2017-07-11 16:36	NaN	0	0	2017-07-11 16:34	
2	1319569	27851	JCG309	2017-07-13 17:33	NaN	0	0	2017-07-13 17:32	
3	1322095	38736	JHH193	2017-07-14 13:02	2017-07-14 13:04	2	0	2017-07-14 13:02	
4	1322398	37449	JHL742	2017-07-14 14:19	2017-07-14 14:20	1	0	2017-07-14 14:19	

In [6]:

```
#Eilučių, su NaN reikšmėmis, ištrynimas
print(len(kelioniu_data))
kelioniu_data = kelioniu_data.dropna()
print(len(kelioniu_data))
```

60834

47927

In [7]:

```
#Dataframe reikšmių reindexavimas
kelioniu_data = kelioniu_data.reset_index(drop=True)
kelioniu_data.head()
```

Out[7]:

	id	customer_id	plate	start	end	duration_min	distance_km	createdat	zone_
0	1322095	38736	JHH193	2017-07-14 13:02	2017-07-14 13:04	2	0	2017-07-14 13:02	
1	1322398	37449	JHL742	2017-07-14 14:19	2017-07-14 14:20	1	0	2017-07-14 14:19	
2	1322771	76526	JFD289	2017-07-14 15:43	2017-07-14 16:36	53	18	2017-07-14 15:41	
3	1323427	76932	JGG686	2017-07-14 18:23	2017-07-14 18:32	9	3	2017-07-14 18:23	
4	1325855	75620	JGJ287	2017-07-15 16:54	2017-07-15 18:02	68	29	2017-07-15 16:49	

In [8]:

```
#Laukų zone_from_id ir zone_to_id tipų suvienodinimas
```

In [9]:

```
kelioniu_data.dtypes
kelioniu_data['zone_to_id'] = kelioniu_data['zone_to_id'].astype('int64')
```

In [10]:

```
kelioniu_data.dtypes
```

Out[10]:

```
id                int64
customer_id       int64
plate             object
start             object
end              object
duration_min      int64
distance_km       object
createdat         object
zone_from_id      int64
zone_to_id        int64
dtype: object
```

In [12]:

```
#Sukuriam du naujus kelionės pradžios ir kelionės pabaigos dataframe.
kelioniu_data['keliones_id'] = kelioniu_data['id']
keliones_start = kelioniu_data[['zone_from_id','plate','start','customer_id','keliones_id']]
keliones_end = kelioniu_data[['zone_to_id','plate','end','customer_id','keliones_id']]
```

In [13]:

```
keliones_start.tail()
```

Out[13]:

	zone_from_id	plate	start	customer_id	keliones_id
47922	28	JGJ228	2017-07-10 10:25	27849	1307624
47923	106	JTR521	2017-07-12 04:55	47738	1313300
47924	106	JHL765	2017-07-12 05:08	47743	1313340
47925	106	JJB577	2017-07-14 04:19	27853	1320295
47926	207	JGH922	2017-07-15 15:32	1834	1325611

In [14]:

```
keliones_end.tail()
```

Out[14]:

	zone_to_id	plate	end	customer_id	keliones_id
47922	1	JGJ228	2017-07-20 16:18	27849	1307624
47923	1	JTR521	2017-07-12 13:55	47738	1313300
47924	1	JHL765	2017-07-12 11:55	47743	1313340
47925	1	JJB577	2017-07-14 15:08	27853	1320295
47926	1	JGH922	2017-07-15 15:39	1834	1325611

In [15]:

```
#Lokacijų duomenis yra sujungiami su keliones_start ir keliones_end duomenimis (taip da
roma norint išvengti geokodavimo
# su visomis reiškėmis)
keliones_start['id'] = keliones_start['zone_from_id']
keliones_end['id'] = keliones_end['zone_to_id']
result_start = pd.merge(keliones_start,
                        data[['id','title','address','x','y']],
                        on='id')
result_start.head()
```

C:\Users\Rokas\Anaconda3\lib\site-packages\ipykernel_launcher.py: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
This is separate from the ipykernel package so we can avoid doing import s until

C:\Users\Rokas\Anaconda3\lib\site-packages\ipykernel_launcher.py: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
after removing the cwd from sys.path.

Out[15]:

	zone_from_id	plate	start	customer_id	keliones_id	id	title	address	x
0	315	JHH193	2017-07-14 13:02	38736	1322095	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981
1	315	JHL742	2017-07-14 14:19	37449	1322398	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981
2	315	JFD289	2017-07-14 15:43	76526	1322771	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981
3	315	JGG686	2017-07-14 18:23	76932	1323427	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981
4	315	JGJ287	2017-07-15 16:54	75620	1325855	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981

In [16]:

```
result_end = pd.merge(keliones_end,
                      data[['id', 'title', 'address', 'x', 'y']],
                      on='id')
result_end.head()
```

Out[16]:

	zone_to_id	plate	end	customer_id	keliones_id	id	title	address	x	
0	315	JHH193	2017-07-14 13:04	38736	1322095	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981	5
1	315	JHL742	2017-07-14 14:20	37449	1322398	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981	5
2	315	JFD289	2017-07-14 16:36	76526	1322771	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981	5
3	315	JGG686	2017-07-14 18:32	76932	1323427	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981	5
4	315	JGJ287	2017-07-15 18:02	75620	1325855	315	Balsiu mokykla - Vilnius	Bubilo g. 8 Vilnius, Lithuania	25.35981	5

In [17]:

```
#Pervadinami laukai, tam kad nesidubliuotų su result start ir result end duomenų lentelės reikšmėmis
result_start['start_x'] = result_start['x']
result_start['start_y'] = result_start['y']
result_end['end_x'] = result_end['x']
result_end['end_y'] = result_end['y']
result_end['address_end'] = result_end['address']
result_end['title_end'] = result_end['title']
```

In [18]:

```
#Kelionės pradžios ir kelionės pabaigos vietas sujungiamos į bendrą dataframe
keliones = pd.merge(result_start,
                    result_end[['keliones_id', 'end', 'zone_to_id', 'address_end', 'title_end', 'end_x', 'end_y']],
                    on='keliones_id')
keliones = keliones[['zone_from_id', 'title', 'address', 'plate', 'start', 'customer_id', 'keliones_id', 'start_x', 'start_y', 'end', 'zone_to_id', 'address_end', 'title_end', 'end_x', 'end_y']]
```


In [19]:

```
#Ištrinamos kelionės kurių pradinis ir galinis taškas yra vienodas (didelė tikimybė, kad tai "nepavykusios" kelionės)
keliones = keliones.loc[(keliones.zone_to_id != keliones.zone_from_id)]
keliones = keliones.sort_values(by='start')
keliones = keliones.reset_index(drop=True)
keliones.head()
```

Out[19]:

	zone_from_id	title	address	plate	start	customer_id	keliones_id	start_x	start_y
0	207	Vilniaus miesto centras - Vilnius	Vienuolio g. 8, Vilnius, Lithuania	JGJ292	2017-01-29 22:06	23998	832310	25.27710	54.6
1	207	Vilniaus miesto centras - Vilnius	Vienuolio g. 8, Vilnius, Lithuania	JHK874	2017-01-29 22:08	35789	832311	25.27710	54.6
2	207	Vilniaus miesto centras - Vilnius	Vienuolio g. 8, Vilnius, Lithuania	JHU889	2017-01-29 22:18	18220	832315	25.27710	54.6
3	207	Vilniaus miesto centras - Vilnius	Vienuolio g. 8, Vilnius, Lithuania	JHU945	2017-01-29 22:22	25070	832320	25.27710	54.6
4	99	PC "Norfa XL", Rygos g. - Vilnius	Rygos g. 13, Vilnius, Lithuania	JHG385	2017-01-29 23:01	39105	832324	25.22552	54.7

In [20]:

```
#keliones start ir keliones end laukai pakeičiami į datos tipo lauką
keliones["start"] = pd.to_datetime(keliones["start"])
keliones["end"] = pd.to_datetime(keliones["end"])
keliones.dtypes
```

Out[20]:

```
zone_from_id      int64
title             object
address           object
plate            object
start            datetime64[ns]
customer_id      int64
keliones_id      int64
start_x          float64
start_y          float64
end              datetime64[ns]
zone_to_id       int64
address_end      object
title_end        object
end_x           float64
end_y           float64
dtype: object
```

In [21]:

```
#Sukuriamas naujos csv failas su kelionėmis
fp = 'C:\Porfolio\citybee_keliones\keliones.csv'
keliones.to_csv(fp)
```

In [22]:

```
#Kelionės sugrupuojamos pagal Lokaciją
keliones_group= keliones.groupby(['start_x', 'start_y']).size().reset_index(name='count
s')
keliones_group.tail()
```

Out[22]:

	start_x	start_y	counts
77	25.319681	54.707518	33
78	25.343030	54.725950	379
79	25.348080	54.785520	37
80	25.359810	54.780100	13
81	25.380210	54.774130	239

In [23]:

```
#Sukuriamas heatmap, kuris parodo užsakymų pasiskirstymą stebėjimo periode
map_heat = folium.Map(location=[54.71378495, 25.2794343068304],
                        tiles = 'CartoDB dark_matter',
                        zoom_start = 12)

heat_data_simple = [[row['start_y'],row['start_x'],row['counts']] for index, row in keliones_group.iterrows()]

HeatMap(heat_data_simple,
        radius = 15
        ).add_to(map_heat)
```

Out[23]:

```
<folium.plugins.heat_map.HeatMap at 0x1f1f8186e80>
```

In [26]:

```
map_heat
```

Out[26]:



In [24]:

```
#Sukuriamas žemėlapiio html failas
map_heat.save('heatmap.html')
```

In [27]:

```
#Sukuriamas naujas laukas kuriame yra atvaizduojamos tik valandos
keliones['start_valanda'] = keliones.start.dt.strftime("%H")
```

In [29]:

```
#Kelionės sugrupuojamos pagal lokaciją ir kelionės pradžios valandą  
keliones_group2= keliones.groupby(['start_x', 'start_y', 'start_valanda']).size().reset_  
index(name='counts')  
keliones_group2.head()
```

Out[29]:

	start_x	start_y	start_valanda	counts
0	25.11011	54.62408	03	3
1	25.11011	54.62408	06	1
2	25.11011	54.62408	07	1
3	25.11011	54.62408	10	1
4	25.11011	54.62408	12	1

In [77]:

```
#Sukuriamas heatmapas kuris parodo kaip skiriasi kelionių užsakymas skirtingu paros metu
map_para = folium.Map(location=[54.71378495, 25.2794343068304],
                        tiles='CartoDB dark_matter',
                        zoom_start = 12,
                        legend_name = 'Number of incidents per district')

heat_df = keliones_group2[['start_y', 'start_x']]

# Sukuriamas weight column, using date
heat_df['Weight'] = keliones_group2['start_valanda']
heat_df['Weight'] = heat_df['Weight'].astype(int)
heat_df = heat_df.dropna(axis=0, subset=['start_y', 'start_x', 'Weight'])

# List comprehension to make out list of lists
heat_data = [[row['start_y'],row['start_x']] for index, row in heat_df[heat_df['Weight'] == i].iterrows()] for i in range(0,24)]

# Plot it on the map
hm = plugins.HeatMapWithTime(heat_data,auto_play=True, radius=20,
                              )
hm.add_to(map_para)
# Display the map
map_para
```

```
C:\Users\Rokas\Anaconda3\lib\site-packages\ipykernel_launcher.py: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

```
if sys.path[0] == '':
```

```
C:\Users\Rokas\Anaconda3\lib\site-packages\ipykernel_launcher.py:13: SettingWithCopyWarning:
```

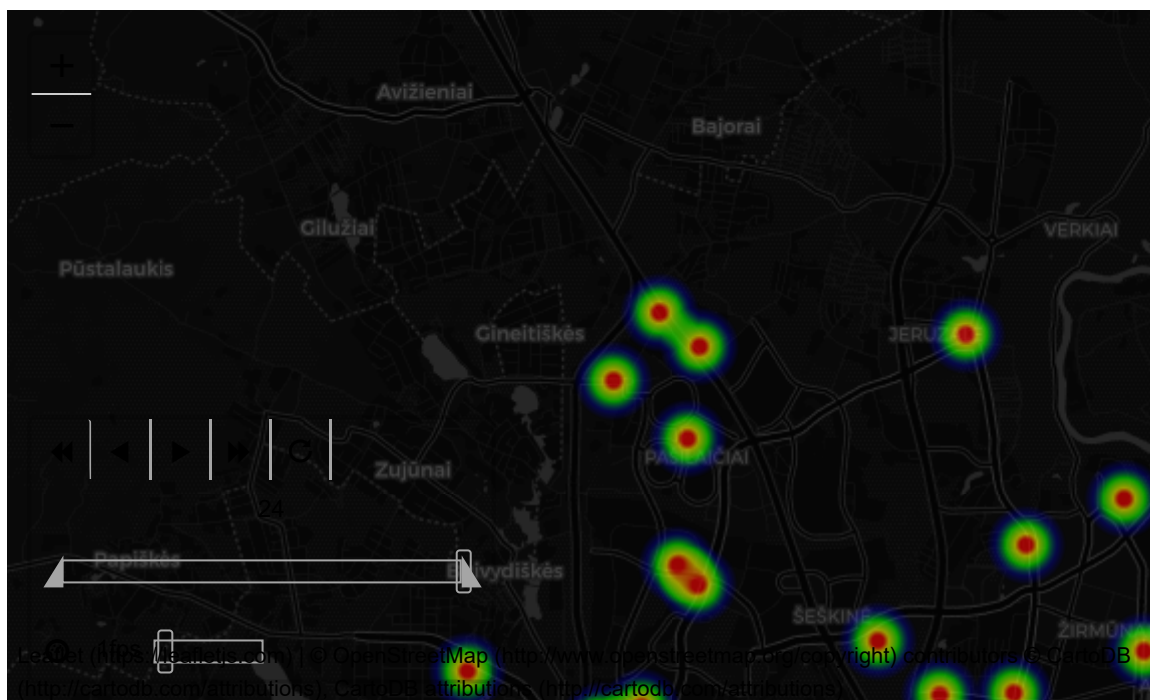
```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
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

```
del sys.path[0]
```

Out[77]:



In [56]:

```
#Sukuriamas žemėlapis html failas
map_para.save('keliones_pagal_laika.html')
```

In [80]:

```
#Sukuriamas vieno automobilio, 7 kelionių maršruto duomenų lentelė
selection2 = keliones.loc[(keliones.plate == 'JHU945')]
selection3 = selection2.head(7)
```

In [85]:

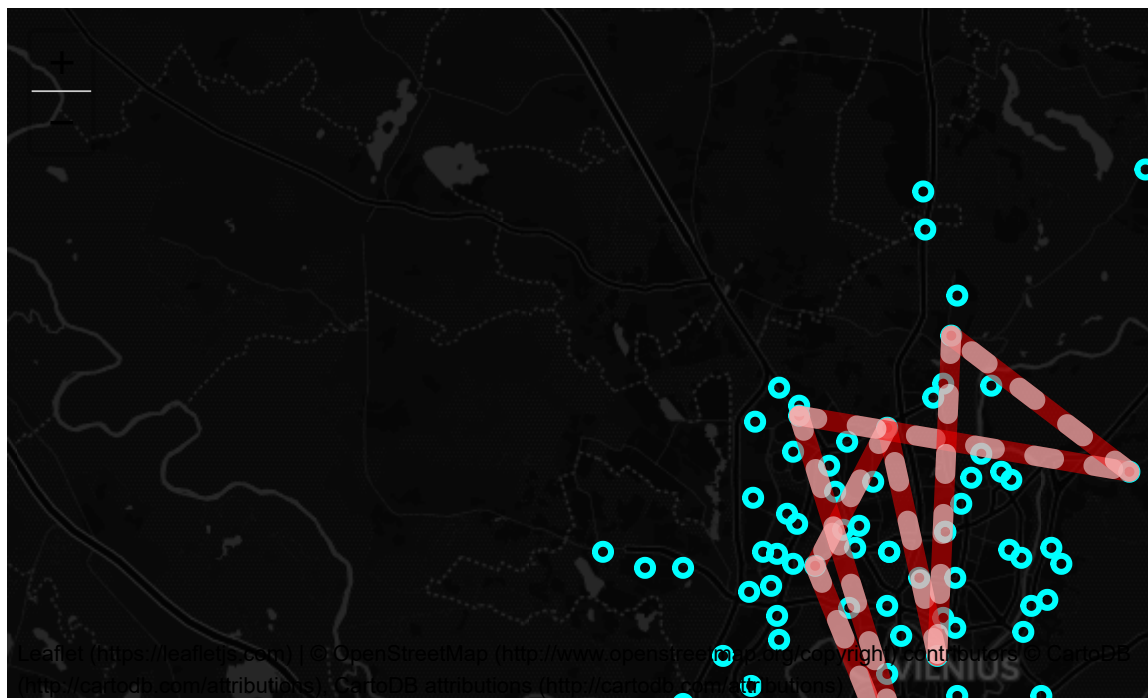
```
#Sukuriamas maršrutas atvaizduojanįs automobilio kelionės maršrutą
path_map = folium.Map(location=[54.71378495, 25.2794343068304],
                        tiles = 'CartoDB dark_matter',
                        zoom_start = 12)

for g in data.itertuples():
    folium.CircleMarker(location=(g.y, g.x),
                        radius=4,
                        popup = folium.Popup(g.title, max_width=500, min_width=200),
                        color='cyan', # divvy color
                        ).add_to(path_map)

selection3.apply(lambda row: folium.plugins.AntPath([(row['start_y'],
                                                         row['start_x']),
                                                         (row['end_y'],
                                                         row['end_x'])]),
                 tooltip = row['address'] + '<br>' + '-->' +
                 row['address_end'] ,
                 color='red', delay = 300, weight=10,
                 ).add_to(path_map),
                axis=1)

path_map
```

Out[85]:



In [84]:

```
#Sukuriamas žemėlapių html failas
path_map.save('automobilio_maršrutas.html')
```