TE360 Final Project

In [1]:
```
import pandas as pd
import plotly.graph_objects as go
from datetime import datetime, date, time
import matplotlib.pyplot as plt
from pandas.plotting import register_matplotlib_converters
import numpy as np
```

Loading datasets

In [2]:
```
congestion_df = pd.read_csv(r'Chicago_Traffic_Tracker_-_Historical_Congestion_Estimates_by_Region_-_2018-Current.csv', low_memory=False)
```

In [3]:
```
congestion_df.head(3)
```

Out[3]:

	TIME	REGION_ID	SPEED	REGION	BUS_COUNT	NUM_READS	HOUR	DAY_OF_WEEK	MONTH	DESCRIPTION	RECORD_ID	WEST	EAST	SOUTH	NORTH	NW_LOCATION	SE_LOCATION
0	03/31/2018 02:40:28 PM	2	27.95	Far North West	25	493	14	7	3	North of Montrose. East River to Cicero	02-201803311940	-87.846210	-87.747456	41.960669	42.019100	POINT (-87.84621 42.0190998)	POINT (-87.747456 41.960669)
1	03/31/2018 02:40:28 PM	3	19.09	North Park-Albany-Linconl Sq	48	834	14	7	3	Montrose to Devon. Cicero to Ravenswood	03-201803311940	-87.747456	-87.674590	41.960669	41.997946	POINT (-87.747456 41.997946)	POINT (-87.67459 41.960669)
2	03/31/2018 02:40:28 PM	4	18.41	Edge Water-Uptown	36	585	14	7	3	Montrose to Devon. Ravenswood to Lake Shore	04-201803311940	-87.674590	-87.646438	41.960669	41.997946	POINT (-87.67459 41.997946)	POINT (-87.646438 41.960669)

In [4]:
```
crash_df = pd.read_csv(r'Traffic_Crashes_-_Crashes.csv', low_memory=False)
crash_df.head(3)
```

Out[4]:

	CRASH_RECORD_ID	RD_NO	CRASH_DATE_EST_I	CRASH_DATE	POSTED_SPEED_LIMIT	TRAFFIC_CONTROL_DEVICE	DEVICE_CONDITION	WEATHER_CONDITION	LIGHTING_CONDITION	FIRST_CRA
0	79c7a2ce89f446262efd86df3d72d18b04ba487024b7c4...	JC199149	NaN	03/25/2019 02:43:00 PM	30	TRAFFIC SIGNAL	FUNCTIONING PROPERLY	CLEAR	DAYLIGHT	
1	792b539deaaad65ee5b4a9691d927a34d298eb33d42af0...	JB422857	NaN	09/05/2018 08:40:00 AM	30	NO CONTROLS	NO CONTROLS	CLEAR	DAYLIGHT	
2	0115ade9a755e835255508463f7e9c4a9a0b47e9304238...	JF318029	NaN	07/15/2022 12:45:00 AM	30	UNKNOWN	UNKNOWN	CLEAR	DARKNESS, LIGHTED ROAD	

3 rows × 49 columns

In [5]:
```
business_df = pd.read_csv(r'Business_Licenses.csv', low_memory=False)
business_df.head(3)
```

Out[5]:

	ID	LICENSE ID	ACCOUNT NUMBER	SITE NUMBER	LEGAL NAME	DOING BUSINESS AS NAME	ADDRESS	CITY	STATE	ZIP CODE	...	LICENSE TERM START DATE	LICENSE TERM EXPIRATION DATE	LICENSE APPROVED FOR ISSUANCE	DATE ISSUED	LICENSE STATUS	LICENSE STATUS CHANGE DATE	SSA	LATITUDE	LONGITUDE	
0	16570-20000216	76522	51755	1	THORNDALE CONSTRUCTION	THORNDALE CONSTRUCTION	11243 CHESAPEAKE PLAC 1ST	WESTCHESTER	IL	60154	...	02/16/2000	02/15/2001	12/22/2003	08/24/2009	AAI	NaN	NaN	NaN	NaN	
1	25710-19960216	119268	52896	1	PAT HAMILTON, INC	PAT HAMILTON CO.	17021 S MAGNOLIA DR 1ST	HAZEL CREST	IL	60429	...	02/16/1996	02/15/1997	12/22/2003	04/12/2006	AAI	NaN	NaN	NaN	NaN	
2	2712733-20210816	2796409	461483	3	J & H DESIGNS INC.	DCI Furniture	2729 W DEVON AVE	CHICAGO	IL	60659	...	08/16/2021	08/15/2023	05/11/2022	05/11/2022	AAI	NaN	43.0	41.997472	-87.698365	(

3 rows × 34 columns

In [6]:

```
police_df = pd.read_csv(r'Police_Stations.csv', low_memory=False)
police_df.head(3)
```

Out[6]:

	DISTRICT	DISTRICT NAME	ADDRESS	CITY	STATE	ZIP	WEBSITE	PHONE	FAX	TTY	X COORDINATE	Y COORDINATE	LATITUDE	LONGITUDE	LOCATION
0	Headquarters	Headquarters	3510 S Michigan Ave	Chicago	IL	60653	http://home.chicagopolice.org	NaN	NaN	NaN	1177731.401	1881697.404	41.830702	-87.623395	(41.8307016873, -87.6233953459)
1	18	Near North	1160 N Larrabee St	Chicago	IL	60610	http://home.chicagopolice.org/community/distri...	312-742-5870	312-742-5771	312-742-5773	1172080.029	1908086.527	41.903242	-87.643352	(41.9032416531, -87.6433521393)
2	19	Town Hall	850 W Addison St	Chicago	IL	60613	http://home.chicagopolice.org/community/distri...	312-744-8320	312-744-4481	312-744-8011	1169730.744	1924160.317	41.947400	-87.651512	(41.9474004564, -87.651512018)

In [7]:

```
region_df = pd.read_csv(r'Chicago_Traffic_Tracker_-_Congestion_Estimates_by_Regions.csv', low_memory=False)
region_df.head(3)
```

Out[7]:

	REGION	REGION_ID	WEST	EAST	SOUTH	NORTH	DESCRIPTION	CURRENT_SPEED	LAST_UPDATED
0	Hyde Park-Kenwood-Woodlawn	21	-87.606334	-87.566260	41.764066	41.822792	71st to Pershing. Cottage Grove to Lake Shore	21.82	2022-11-16 16:11:58.0
1	Downtown Lakefront	29	-87.623080	-87.595378	41.866129	41.911401	Roosevelt to Oak. Michigan to Lake Shore	19.77	2022-11-16 16:11:58.0
2	Lincoln Park-Lake View	8	-87.674590	-87.619112	41.910561	41.960669	North Ave to Montrose. Ravenswood to Lake Shore	20.45	2022-11-16 16:11:58.0

In [8]:

```
print(len(congestion_df["REGION_ID"].unique()))
print(len(congestion_df["REGION"].unique()))
print(len(congestion_df["DESCRIPTION"].unique()))
print(len(congestion_df["DAY_OF_WEEK"].unique()))
```

```
29
29
29
7
```

In [9]:

```
id_info = {}

for index, row in region_df.iterrows():
    loc_list = [row[' WEST'], row[' EAST'], row[' SOUTH'], row[' NORTH']]
    temp_dict = {
        "loc" : loc_list,
        "reg" : row[' REGION_ID'],
        "desc" : row[' DESCRIPTION'],
    }
    id_info[row[' REGION_ID']] = temp_dict
```

In [10]:

```
print("congestion_df\n", congestion_df.isnull().sum().sort_values())
print("business_df\n", business_df.isnull().sum().sort_values())
print("police_df\n", police_df.isnull().sum().sort_values())
print("crash_df\n", crash_df.isnull().sum().sort_values())
```

```
congestion_df
 TIME           0
NORTH          0
SOUTH          0
EAST           0
WEST           0
RECORD_ID      0
DESCRIPTION    0
NW_LOCATION    0
MONTH          0
HOUR           0
NUM_READS      0
BUS_COUNT      0
REGION         0
SPEED          0
REGION_ID      0
DAY_OF_WEEK    0
SE_LOCATION    0
dtype: int64
business_df
 ID                                          0
LICENSE ID                                  0
ACCOUNT NUMBER                              0
SITE NUMBER                                 0
LEGAL NAME                                  0
LICENSE DESCRIPTION                         0
ADDRESS                                     0
LICENSE CODE                                0
DATE ISSUED                                 0
LICENSE STATUS                              0
CONDITIONAL APPROVAL                        0
APPLICATION TYPE                            0
LICENSE NUMBER                              1
CITY                                        3
STATE                                      12
LICENSE TERM EXPIRATION DATE              123
DOING BUSINESS AS NAME                    167
ZIP CODE                                  285
LICENSE TERM START DATE                  2701
APPLICATION REQUIREMENTS COMPLETE       18351
PAYMENT DATE                            22291
LICENSE APPROVED FOR ISSUANCE           51362
WARD PRECINCT                           80729
WARD                                    80755
LATITUDE                                86301
LOCATION                                86301
LONGITUDE                               86301
POLICE DISTRICT                        126485
PRECINCT                               132705
BUSINESS ACTIVITY                      541848
BUSINESS ACTIVITY ID                   541848
SSA                                    804150
APPLICATION CREATED DATE               832987
LICENSE STATUS CHANGE DATE            1012058
dtype: int64
police_df
 DISTRICT         0
DISTRICT NAME    0
ADDRESS          0
CITY             0
STATE            0
ZIP              0
WEBSITE          0
X COORDINATE     0
Y COORDINATE     0
LATITUDE         0
LONGITUDE        0
LOCATION         0
PHONE            1
FAX              1
TTY              1
```

```
dtype: int64
crash_df
 CRASH_RECORD_ID                        0
CRASH_MONTH                            0
CRASH_DAY_OF_WEEK                      0
CRASH_HOUR                             0
NUM_UNITS                              0
SEC_CONTRIBUTORY_CAUSE                 0
PRIM_CONTRIBUTORY_CAUSE                0
DATE_POLICE_NOTIFIED                   0
DAMAGE                                 0
CRASH_TYPE                             0
ROAD_DEFECT                            0
ROADWAY_SURFACE_COND                   0
STREET_NO                              0
TRAFFICWAY_TYPE                        0
FIRST_CRASH_TYPE                       0
POSTED_SPEED_LIMIT                     0
LIGHTING_CONDITION                     0
WEATHER_CONDITION                      0
DEVICE_CONDITION                       0
TRAFFIC_CONTROL_DEVICE                 0
ALIGNMENT                              0
CRASH_DATE                             0
STREET_NAME                            1
STREET_DIRECTION                       4
BEAT_OF_OCCURRENCE                     5
INJURIES_TOTAL                      1451
INJURIES_NON_INCAPACITATING         1451
INJURIES_REPORTED_NOT_EVIDENT       1451
INJURIES_NO_INDICATION              1451
INJURIES_INCAPACITATING             1451
INJURIES_FATAL                      1451
INJURIES_UNKNOWN                    1451
MOST_SEVERE_INJURY                  1462
RD_NO                               4220
LONGITUDE                           4230
LATITUDE                            4230
LOCATION                            4230
REPORT_TYPE                        18180
HIT_AND_RUN_I                     465042
LANE_CNT                          474308
INTERSECTION_RELATED_I            519134
CRASH_DATE_EST_I                  622301
NOT_RIGHT_OF_WAY_I                641693
STATEMENTS_TAKEN_I                659317
PHOTOS_TAKEN_I                    664992
WORK_ZONE_I                       669330
WORK_ZONE_TYPE                    670183
DOORING_I                         671198
WORKERS_PRESENT_I                 672275
dtype: int64
```

Cleaning and simplifying dataset

In [11]:

```
congestion_df_sub = congestion_df[['TIME', 'DAY_OF_WEEK', 'REGION_ID', 'SPEED', 'REGION', 'DESCRIPTION']]
print(congestion_df_sub.isnull().sum().sort_values())
print(len(congestion_df_sub))
```

```
TIME           0
DAY_OF_WEEK    0
REGION_ID      0
SPEED          0
REGION         0
DESCRIPTION    0
dtype: int64
6530484
```

In [12]:

```
business_df_sub = business_df[["ID", "LATITUDE", "LONGITUDE"]].reset_index(drop=True).dropna(axis=0)
print(business_df_sub.isnull().sum().sort_values())
print(len(business_df_sub))
```

```
ID           0
LATITUDE     0
LONGITUDE    0
dtype: int64
999096
```

In [13]:

```
police_df_sub = police_df[["DISTRICT", "LATITUDE", "LONGITUDE"]].reset_index(drop=True)
print(police_df_sub.isnull().sum().sort_values())
print(len(police_df_sub))
```

```
DISTRICT     0
LATITUDE     0
LONGITUDE    0
dtype: int64
23
```

In [14]:

```
crash_df_sub = crash_df[["CRASH_DATE", "CRASH_DAY_OF_WEEK", "LATITUDE", "LONGITUDE"]].reset_index(drop=True).dropna(axis=0)
print(crash_df_sub.isnull().sum().sort_values())
print(len(crash_df_sub))
```

```
CRASH_DATE           0
CRASH_DAY_OF_WEEK    0
LATITUDE             0
LONGITUDE            0
dtype: int64
669073
```

Attach region info to business, police and crash

In [15]:

```
def calc_region(lat, lon):
    for key, value in id_info.items():
        if value["loc"][0] <= lon <= value["loc"][1] and value["loc"][2] <= lat <= value["loc"][3]:
            return key
    return -1
```

In [16]:

```
def attach_region(df):
    df.reset_index()
    temp_list = []
    for index, row in df.iterrows():
#         print("curr row", row)
        temp_list.append(calc_region(row["LATITUDE"], row["LONGITUDE"]))
#         print("done")
#     print(temp_list)
    df["REGION_ID"] = temp_list
```

In [17]:

```
attach_region(police_df_sub)
police_df_sub
```

Out[17]:

	DISTRICT	LATITUDE	LONGITUDE	REGION_ID
0	Headquarters	41.830702	-87.623395	16
1	18	41.903242	-87.643352	12
2	19	41.947400	-87.651512	8
3	20	41.979550	-87.692845	3
4	22	41.691435	-87.668520	25
5	24	41.999763	-87.671324	1
6	25	41.918609	-87.765574	5
7	1	41.858373	-87.627356	16
8	2	41.801811	-87.630560	20
9	3	41.766431	-87.605748	21
10	4	41.707933	-87.568349	27
11	5	41.692723	-87.604506	26
12	6	41.752137	-87.644229	23
13	7	41.779632	-87.660887	19
14	8	41.778987	-87.708864	18
15	9	41.837394	-87.646408	15
16	10	41.856685	-87.708382	14
17	11	41.873582	-87.705488	10
18	12	41.862977	-87.656973	15
19	14	41.921103	-87.697452	7
20	15	41.880083	-87.768200	9
21	16	41.974094	-87.766149	2
22	17	41.966053	-87.728115	3

In [18]:

```
attach_region(business_df_sub)
business_df_sub
```

Out[18]:

	ID	LATITUDE	LONGITUDE	REGION_ID
2	2712733-20210816	41.997472	-87.698365	3
33	2617269-20220916	41.820285	-87.665397	19
34	2652692-20201217	42.019413	-87.688628	1
35	2647849-20200916	41.885945	-87.653462	11
36	1579838-20220616	41.961655	-87.655650	4
...	...	...	...	...
1085392	2574974-20220416	41.905559	-87.632008	12
1085393	2767340-20230116	41.950396	-87.742151	6
1085394	2881783-20221121	41.883066	-87.615456	29
1085395	2506759-20230116	41.980323	-87.692610	3
1085396	2215982-20221016	41.775718	-87.644885	19

999096 rows × 4 columns

In [19]:

```
attach_region(crash_df_sub)
```

```
crash_df_sub
```

Out[19]:

	CRASH_DATE	CRASH_DAY_OF_WEEK	LATITUDE	LONGITUDE	REGION_ID
0	03/25/2019 02:43:00 PM	2	41.884547	-87.641201	13
1	09/05/2018 08:40:00 AM	4	41.968562	-87.740659	3
2	07/15/2022 12:45:00 AM	6	41.886336	-87.716203	10
3	08/29/2022 11:30:00 AM	2	41.749348	-87.721097	22
4	07/15/2022 06:50:00 PM	6	41.925111	-87.667997	8
...	...	...	...	...	...
673298	11/24/2022 04:40:00 PM	5	41.821258	-87.701873	18
673299	11/24/2022 03:50:00 PM	5	41.686263	-87.664858	25
673300	11/24/2022 01:00:00 PM	5	41.985267	-87.662517	4
673301	11/24/2022 06:47:00 AM	5	41.874644	-87.629649	13
673302	11/22/2022 04:30:00 PM	3	41.871449	-87.685432	11

669073 rows × 5 columns

Seperate into weekday and weekend

In [20]:

```
congestion_df_sub_week = congestion_df_sub[(congestion_df_sub['DAY_OF_WEEK'] != 7) & (congestion_df_sub['DAY_OF_WEEK'] != 1)]
congestion_df_sub_week = congestion_df_sub_week[["SPEED", "REGION_ID"]]
congestion_df_sub_week = congestion_df_sub_week.groupby(['REGION_ID']).mean()
congestion_df_sub_week
```

Out[20]:

	SPEED
REGION_ID	
1	20.421086
2	26.658373
3	22.737499
4	19.712142
5	22.380799
6	21.892023
7	21.021016
8	20.661609
9	22.143575
10	23.050108
11	22.313436
12	18.847322
13	18.672129
14	23.250800
15	25.231502
16	24.195353
17	24.497953
18	24.590779
19	25.543682
20	25.269371
21	23.133598
22	27.395769
23	23.491698
24	24.334073
25	25.284511
26	28.198309
27	27.766483
28	21.051773
29	19.249403

In [21]:

```
crash_df_sub_count = crash_df_sub[(crash_df_sub['CRASH_DAY_OF_WEEK'] != 7) & (crash_df_sub['CRASH_DAY_OF_WEEK'] != 1) & (crash_df_sub['REGION_ID'] != -1)]
crash_df_sub_count = crash_df_sub_count[["REGION_ID"]]
crash_df_sub_count = crash_df_sub_count.groupby(['REGION_ID']).size().reset_index(name='COUNT')
crash_df_sub_count
```

Out[21]:

	REGION_ID	COUNT
0	1	10656
1	2	15910
2	3	20959
3	4	12132
4	5	25161
5	6	19575
6	7	18677
7	8	22808
8	9	13044
9	10	28857
10	11	27306
11	12	15327
12	13	22114
13	14	19389
14	15	16239
15	16	11298
16	17	5675
17	18	33180
18	19	19096
19	20	16070
20	21	15356
21	22	9939
22	23	21980
23	24	16704
24	25	8414
25	26	17878
26	27	6144
27	28	4516
28	29	9714

In [22]:

```
business_df_sub_count = business_df_sub[business_df_sub['REGION_ID'] != -1]
business_df_sub_count = business_df_sub_count[["REGION_ID"]]
business_df_sub_count = business_df_sub_count.groupby(['REGION_ID']).size().reset_index(name='COUNT')
business_df_sub_count
```

Out[22]:

	REGION_ID	COUNT
0	1	20902
1	2	32401
2	3	57885
3	4	30462
4	5	51615
5	6	51130
6	7	46028
7	8	82740
8	9	11594
9	10	34065
10	11	69060
11	12	61331
12	13	100289
13	14	36096
14	15	38618
15	16	19177
16	17	11185
17	18	47789
18	19	31763
19	20	14432
20	21	16960
21	22	13250
22	23	21658
23	24	23983
24	25	13411
25	26	20368
26	27	9679
27	28	5259
28	29	18187

Showing regions on map

In [23]:

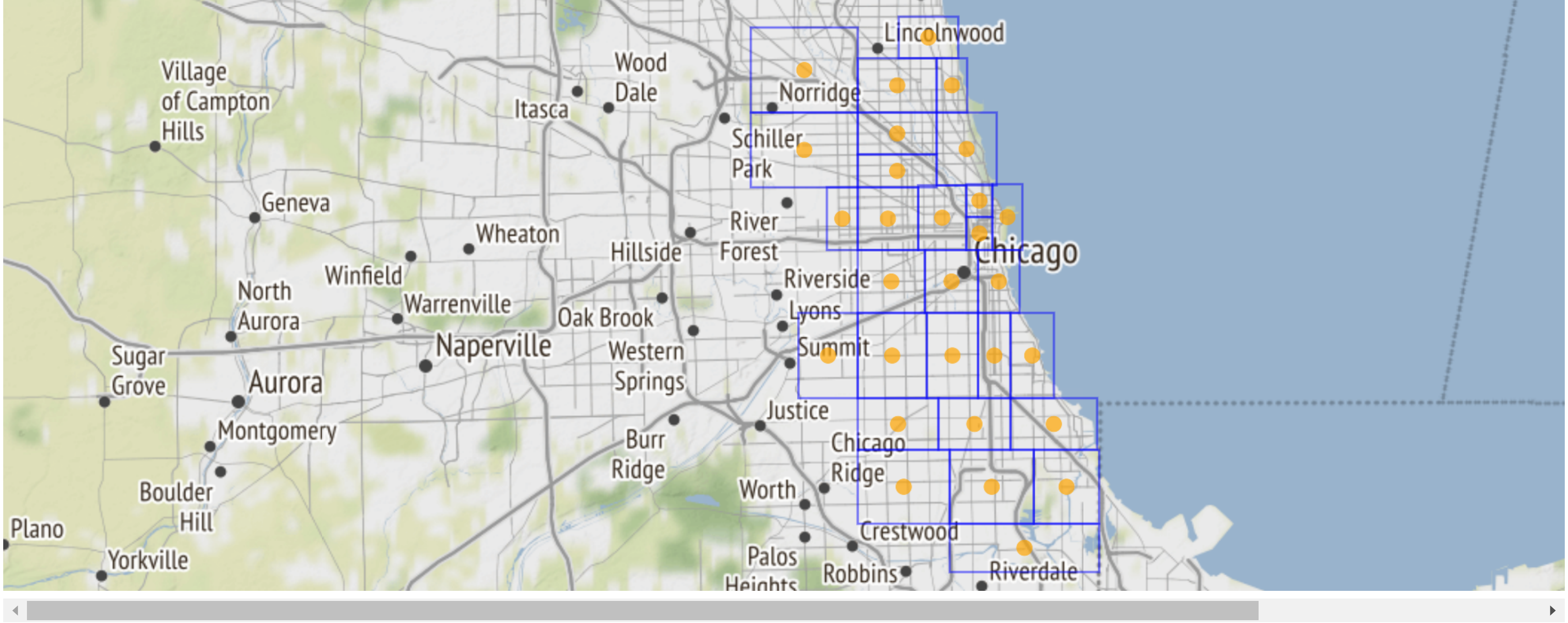
```
def get_region_coord(id_info_dict):
    coord = []
    center_lon = []
    center_lat = []
    text = []
    for key, value in id_info.items():
        row[' WEST'], row[' EAST'], row[' SOUTH'], row[' NORTH']
        w = value["loc"][0]
        e = value["loc"][1]
        s = value["loc"][2]
        n = value["loc"][3]
        coord.append([[[w, n], [w, s], [e, s], [e, n] ,[w, n]]])
        center_lon.append((w+e)/2)
        center_lat.append((n+s)/2)
        text.append("Region #"+str(key)+": "+value["desc"])
    return coord, center_lon, center_lat, text
```

In [24]:

```
coord, center_lon, center_lat, text = get_region_coord(id_info)

speed_list = congestion_df_sub_week["SPEED"].reset_index()["SPEED"]
crash_list = crash_df_sub_count["COUNT"]
business_list = business_df_sub_count["COUNT"]

final_text = res = [i+". "+"Speed:"+f'{j:.2f}'+". "+"Crashes:"+str(k)+". "+"Businesses:"+str(l)+". " for i, j, k, l in zip(text, congestion_df_sub_week["SPEED"], crash_df_sub_count["COUNT"], bu
```

In [25]:

```
fig = go.Figure(go.Scattermapbox(
        lat=center_lat,
        lon=center_lon,
        mode='markers',
        marker=go.scattermapbox.Marker(
            size=15,
            color='orange',
            opacity=0.7
        ),
        text=final_text,
    ))


fig.update_layout(
    mapbox = {
        'style': "stamen-terrain",
        'center': {
            'lon': -87.65,
            'lat': 41.835
        },
        'zoom': 9.4,
        'layers': [{
            'source': {
                'type': "FeatureCollection",
                'features': [{
                    'type': "Feature",
                    'geometry': {
                        'type': "MultiPolygon",
                        'coordinates': coord,
                    }
                }]
            },
            'type': "line",
            'below': "traces",
            'color': "blue",
            'opacity': 0.5
        }]},
    margin = {'l':0, 'r':1, 'b':0, 't':0})
# fig.update_traces(cluster=dict(enabled=True))
fig.show()
```



Correlation

```
In [26]: df_all = pd.DataFrame({'Business':business_list, 'Speed': speed_list, 'Crash': crash_list})
         df_all
```

Out[26]:

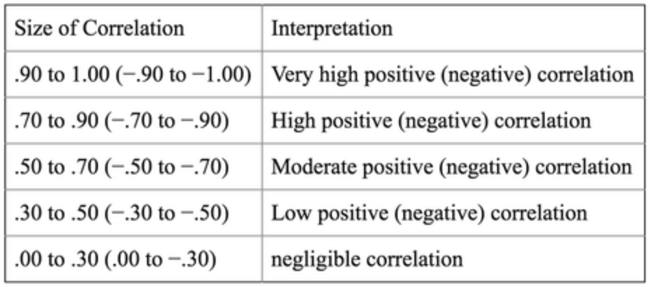
	Business	Speed	Crash
0	20902	20.421086	10656
1	32401	26.658373	15910
2	57885	22.737499	20959
3	30462	19.712142	12132
4	51615	22.380799	25161
5	51130	21.892023	19575
6	46028	21.021016	18677
7	82740	20.661609	22808
8	11594	22.143575	13044
9	34065	23.050108	28857
10	69060	22.313436	27306
11	61331	18.847322	15327
12	100289	18.672129	22114
13	36096	23.250800	19389
14	38618	25.231502	16239
15	19177	24.195353	11298
16	11185	24.497953	5675
17	47789	24.590779	33180
18	31763	25.543682	19096
19	14432	25.269371	16070
20	16960	23.133598	15356
21	13250	27.395769	9939
22	21658	23.491698	21980
23	23983	24.334073	16704
24	13411	25.284511	8414
25	20368	28.198309	17878
26	9679	27.766483	6144
27	5259	21.051773	4516
28	18187	19.249403	9714

In [27]:

```
df_all.corr()
```

Out[27]:

	Business	Speed	Crash
Business	1.000000	-0.486493	0.656232
Speed	-0.486493	1.000000	-0.104016
Crash	0.656232	-0.104016	1.000000



Size of Correlation	Interpretation
.90 to 1.00 (−.90 to −1.00)	Very high positive (negative) correlation
.70 to .90 (−.70 to −.90)	High positive (negative) correlation
.50 to .70 (−.50 to −.70)	Moderate positive (negative) correlation
.30 to .50 (−.30 to −.50)	Low positive (negative) correlation
.00 to .30 (.00 to −.30)	negligible correlation

https://towardsdatascience.com/eveything-you-need-to-know-about-interpreting-correlations-2c485841c0b8

Polynomial regression prediction

In [28]:

```
congestion_df = pd.read_csv(r'Chicago_Traffic_Tracker_-_Historical_Congestion_Estimates_by_Region_-_2018-Current.csv', low_memory=False)
congestion_df_sub = congestion_df[['TIME', 'DAY_OF_WEEK', 'REGION_ID', 'SPEED', 'REGION', 'DESCRIPTION']]
congestion_df_home = congestion_df_sub[(congestion_df_sub['DAY_OF_WEEK'] < 7) & (1 < congestion_df_sub['DAY_OF_WEEK']) & (9 <= congestion_df_sub['REGION_ID']) & (congestion_df_sub['REGION_ID']
congestion_df_home = congestion_df_home[["TIME", "SPEED", 'REGION_ID']]
```

In [29]:

```
congestion_df_home['TIME'] = pd.to_datetime(congestion_df_home['TIME'])
```

In [30]:

```
congestion_df_mean = congestion_df_home.copy().dropna()
congestion_df_mean = congestion_df_mean.resample('10min', on='TIME').mean().dropna()
congestion_df_mean = congestion_df_mean.reset_index()
congestion_df_mean["TIME"] = congestion_df_mean["TIME"].apply(lambda a : a.time())
congestion_df_mean = congestion_df_mean.groupby('TIME').mean()
congestion_df_mean = congestion_df_mean.reset_index()
congestion_df_mean
```

Out[30]:

	TIME	SPEED	REGION_ID
0	00:00:00	25.809224	10.75
1	00:10:00	25.864250	10.75
2	00:20:00	25.999773	10.75
3	00:30:00	26.497598	10.75
4	00:40:00	26.102901	10.75
...	...	...	...
139	23:10:00	24.656831	10.75
140	23:20:00	24.789538	10.75
141	23:30:00	25.072350	10.75
142	23:40:00	25.176323	10.75
143	23:50:00	25.456009	10.75

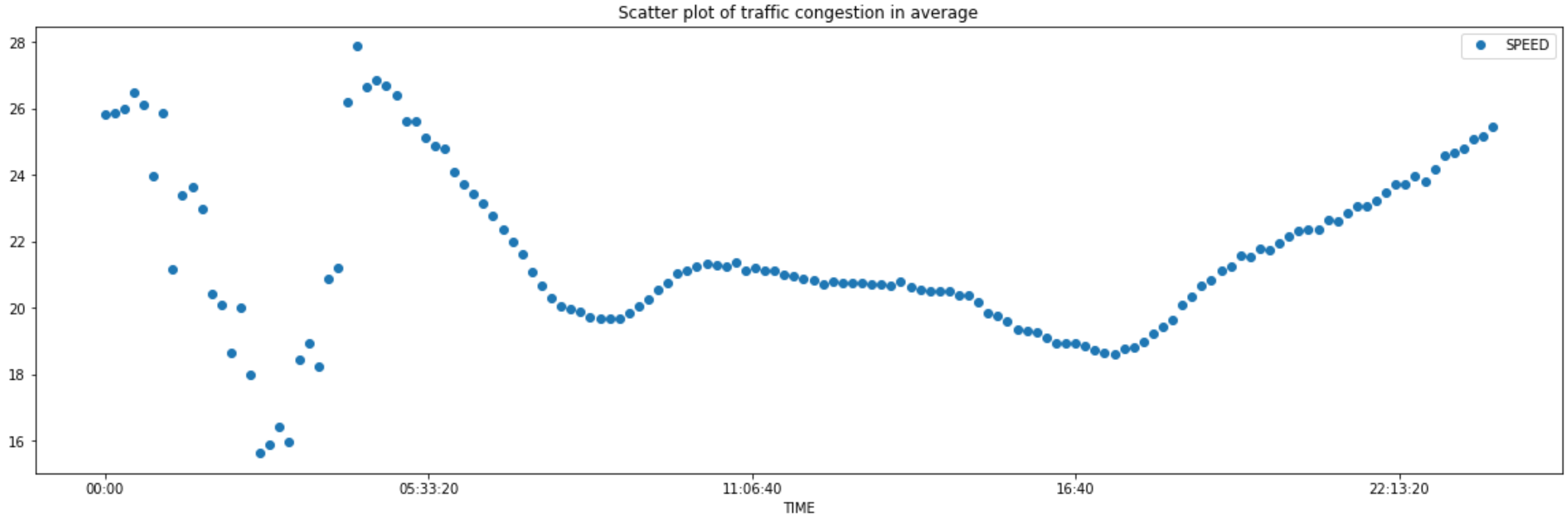
144 rows × 3 columns

In [31]:

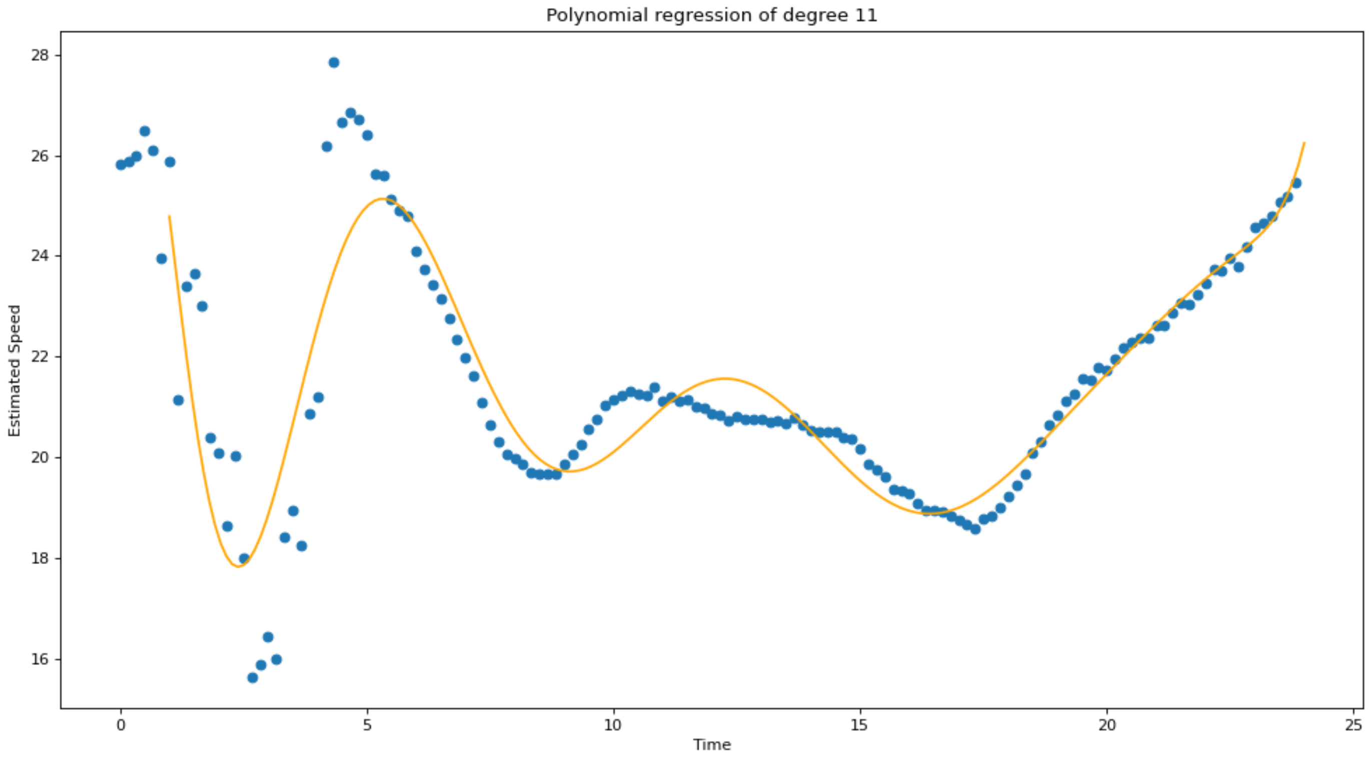
```
time_list = congestion_df_mean["TIME"].apply(lambda a:a.hour+a.minute/60.0).tolist()
speed_mean_list = congestion_df_mean["SPEED"].tolist()

congestion_df_mean.plot(x='TIME', y='SPEED', style='o', figsize=(20, 6), title="Scatter plot of traffic congestion in average")
```

Out[31]:

```
<AxesSubplot:title={'center':'Scatter plot of traffic congestion in average'}, xlabel='TIME'>
```



```
In [32]: mymodel = np.poly1d(np.polyfit(time_list, speed_mean_list, 11))
         myline = np.linspace(1, 24, 200)

         plt.figure(figsize=(15, 8), dpi=80)
         plt.scatter(time_list, speed_mean_list)
         plt.plot(myline, mymodel(myline), color="orange")
         plt.title("Polynomial regression of degree 11")
         plt.xlabel("Time")
         plt.ylabel("Estimated Speed")
         plt.show()
```



```
In [33]: mymodel(5)

Out[33]: 24.9773136119226

In [34]: mymodel(17)

Out[34]: 18.993878606186545

In [ ]:
```