

Chicago Food Inspections

Recent watchdog report published by [Chicago Tribune](#) indicated that food safety inspectors overlook hundreds of day cares in the city of Chicago.



The key take away from the Chicago Tribune watchdog report is that the city had only 33 working field inspectors to cover the entire city of Chicago. Many of the facilities serve food for Children, and while few fail inspections, many escape routine inspections.

Here the goal is to identify the **hot-spots** (areas that have facilities serving food to children and have failed inspections in the past) on the Chicago map to dispatch inspectors to.

To achieve this goal, these are the necessary steps:

1. Dataset for Chicago Food Inspections
2. NoSQL database Engine (ElasticSearch) for indexing and data retrieval
3. HeatMap to plot the children facilities that failed Chicago Food Inspections

Loading the Dataset CSV file

Description of the dataset:

1. It has 164953 inspection records
2. It has inspection records from 2010 to 2018
3. It has 17 fields

```
In [1]: # Load the CSV Chicago Food Inspections dataset into a dataframe
import pandas as pd
import numpy as np

df = pd.read_csv("C:\\\\Users\\\\richd\\\\420\\\\Food_Inspections.csv")
```

In [2]: df.head()

Out[2]:

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	St
0	2561538	PACINO'S RC, LLC	PACINOS	2683667.0	Restaurant	Risk 1 (High)	1010 S DELANO CT	CHICAGO	
1	2556917	BISTRO	BISTRO	2846045.0	Restaurant	Risk 1 (High)	1400 S JEAN BAPTISTE POINTE DUSABLE LAKESHORE DR	CHICAGO	
2	2549763	DANIEL WILLIAM HALE	DANIEL WILLIAM HALE	66311.0	School	Risk 1 (High)	4934 S Wabash (45E) (C/Shabazz)	CHICAGO	
3	2492473	DANIEL WILLIAM HALE	DANIEL WILLIAM HALE	66311.0	School	Risk 1 (High)	4934 S Wabash (45E) (C/Shabazz)	CHICAGO	
4	2384228	YOUTH CONNECTION	YOUTH CONNECTION LEADERSHIP ACADEMY	66101.0	School	Risk 2 (Medium)	3424 S State (1E&W) ST	CHICAGO	

There are few fields in the dataset of interest:

1. Risk
2. Results
3. Latitude
4. Longitude
5. Inspection ID

There are possibilities that the data entry clerk might've made some typos and misspellings and there are different words meant to indicate the same thing, some examples of this:

- Children
- Children's
- Childrens

In [83]: #Import Elasticsearch and helpers from elasticsearch

```
from elasticsearch import Elasticsearch, helpers
```

```
es=Elasticsearch('http://student:spsdata@129.105.248.25:9200')
```

Load and Index the Inspection Records into ElasticSearch

Inspection records are inserted into ElasticSearch engine using the bulk API of elastic search.

Here is the link [API DOCS](#) for the API documentation.

Query is used to retrieve data from ElasticSearch server

The query is used to retrieve data from ElasticSearch servers that match certain filters.

Three main experiments and will compare results for each:

1. Experiment #1: Using Regular Expressions using the term Children
2. Experiment #2: Using Fuzziness using the term Children's
3. Experiment #3: Using Fuzziness using the term Children

Experiment #1: Create the query using regex

```
In [4]: query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{"match" : {"Results": "Fail"}}, {"match" : {"Risk": "Critical"}}, {"match" : {"Facility Type": "Food Service Establishment"}}, {"match" : {"Violations": "Yes"}}, {"match": {"Address": "Children*"}, "#using regex of children's name"}], "should": [{"match": {"Address": "Children*"}, "#using regex of children's name"}]}
        }
    }
results = es.search(index='food_inspections', body=query, scroll='1h')
```

```
In [5]: sid = results['_scroll_id']
scroll_size = results['hits']['total']

print('sid = ', sid)
print('Scroll Size = ', scroll_size)
```

```
sid = DnF1ZXJ5VGh1bkZ1dGNoCgAAAAAD1EWVFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFmBZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPURZYWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAAD1EWXFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFmxZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPURZkWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAAD1EWcFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0E=
Scroll Size = 601
```

```
In [6]: type(results)
```

Out[6]: dict

Process the retrieved documents and filter fields for the Heatmap

Create a list-of-lists of the two fields, (Latitude and Longitude) for the HeatMap

In [9]: len(results['hits']['hits'])

Out[9]: 601

```
In [10]: count = 0
list_of_LAT_LONG_pairs = []
while(scroll_size > 0):

    for inspection in results['hits']['hits']:                      #Iterating each result
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None):
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
                        list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

    results = es.scroll(scroll_id = sid, scroll = '2m')           #Changing the scroll-id
    sid = results['_scroll_id']
    scroll_size = len(results['hits']['hits'])

print("the total number of match with children using wild card:",count)
```

the total number of match with children using wild card: 601

In [11]: document.keys()

Out[11]: dict_keys(['Inspection ID', 'DBA Name', 'AKA Name', 'License #', 'Facility Type', 'Risk', 'Address', 'City', 'State', 'Zip', 'Inspection Date', 'Inspection Type', 'Results', 'Violations', 'Latitude', 'Longitude', 'Location'])

In [12]: list_of_LAT_LONG_pairs[:3]

Out[12]: [[41.8814369069, -87.6659213595],
[41.760441801, -87.6735652436],
[41.9531127244, -87.7800185741]]

In [13]: len(list_of_LAT_LONG_pairs)

Out[13]: 601

Install folium package to plot the Map and Heatmaps

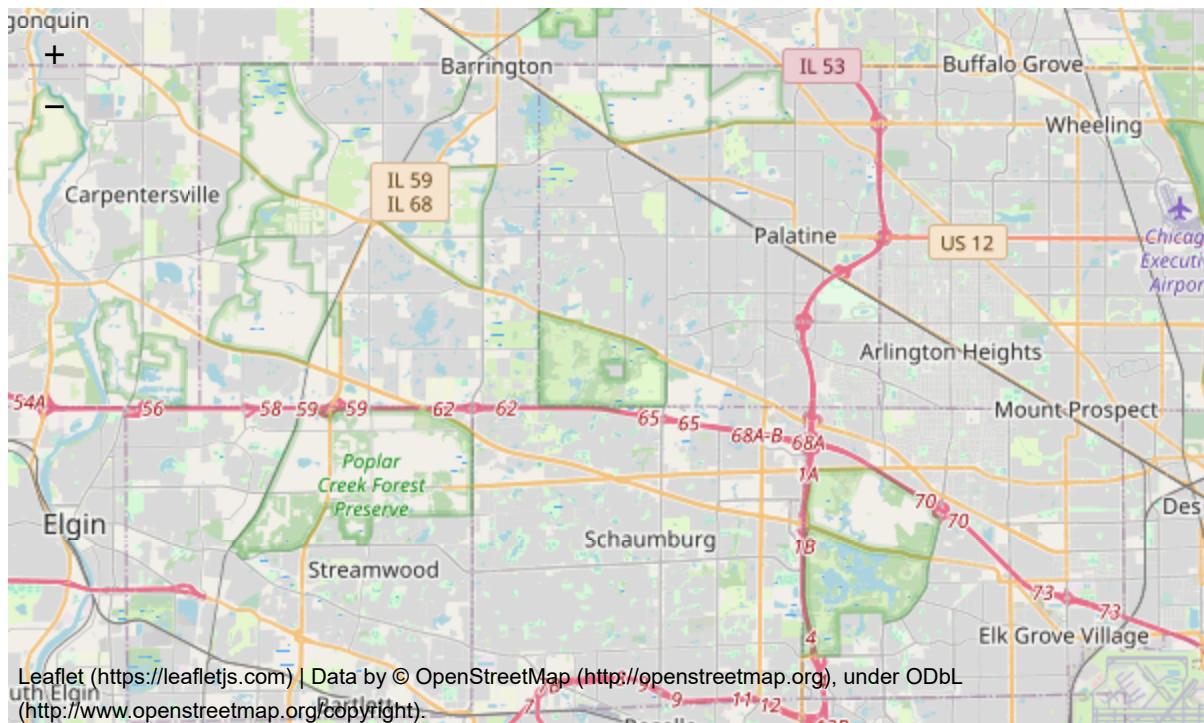
```
In [14]: import folium
from folium import plugins

print(folium.__version__)

0.13.0
```

```
In [15]: chicago_map = folium.Map([41.90293279, -87.70769386], zoom_start=11)
chicago_map
```

Out[15]:

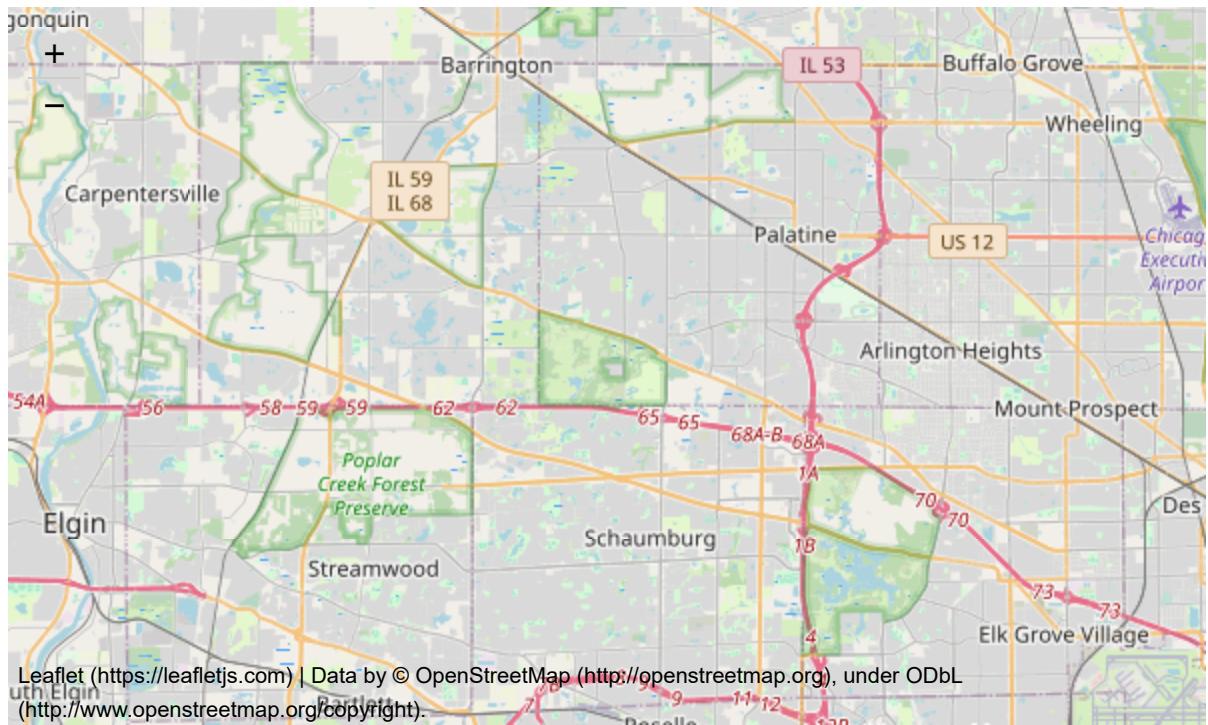


Create the HeatMap

```
In [16]: # Plot the query matches on Chicago HeatMap

chicago_map.add_child(plugins.HeatMap(list_of_LAT_LONG_pairs, radius=15))
chicago_map
```

Out[16]:



Create the query using fuzziness

Experiment #2: Run a fuzzy query using the ~ operator and the term Children:

- "query": "Children~2"

In [17]:

```
query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{}{'match' : {'Results': 'Fail'}}, {"match" : {'Risk': {"operator": "gt", "value": 10}}}],
            "query_string": {
                "query": "Children~2",
                "fields": ["Facility Type", "Violations", "Dishware"]
            }
        }
    }
}
results = es.search(index='food_inspections', body=query, scroll='1h')
```

In [19]:

```
sid = results['_scroll_id']
scroll_size = results['hits']['total']
```

In [20]:

```
print('sid = ', sid)
print('Scroll Size = ', scroll_size)
```

```

sid = DnF1ZXJ5VGh1bkZldGNoCgAAAAAD1EWoFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFpxZsUk
40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPURaYWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EWkFmxSTjR
3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFpRZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPURakWbFJONHdM
UGxRei1DZ1FDS2hjOUZDQQAAAAD1EWrFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFrBZsUk40d0xQb
FF6LUNmUUNLaGM5RkNBAAAAAPURaoWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EWtFmxSTjR3TFBsUX
otQ2ZRQ0toYz1GQ0E=
Scroll Size = 602

```

```

In [21]: count = 0
list_of_LAT_LONG_pairs = []

while(scroll_size > 0):

    for inspection in results['hits']['hits']:
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None):
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
                        list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

    results = es.scroll(scroll_id = sid, scroll = '2m')
    sid = results['_scroll_id']
    scroll_size = len(results['hits']['hits'])

print("Total number of query matches with children using fuzziness:",count)

```

Total number of query matches with children using fuzziness: 602

Experiment #3: Run a fuzzy query using the ~ operator and the term Children's :

- "query": "Children's~2"

```

In [48]: query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{"match" : {"Results": 'Fail'}}, {"match" : {"Risk": "D"}}

            {"query_string": {
                "query": "Children's~2",
                "fields": ["Facility Type", "Violations", "D"]
            }}
        }
    }
}
results = es.search(index='food_inspections', body=query, scroll='1h')

```

```
In [24]: sid = results['_scroll_id']
scroll_size = results['hits']['total']
```

```
In [25]: print('sid = ', sid)
print('Scroll Size = ', scroll_size)
```

```
sid = DnF1ZXJ5VGhbkZldGNoCgAAAAAD1EWvFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFrhZsUk
40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPURbAWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EWyFmxSTjR
3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFsRZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPURbYWbFJONHdM
UGxRei1DZ1FDS2hjOUZDQQAAAAD1EWzFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RFtBzsUk40d0xQb
FF6LUNmUUNLaGM5RkNBAAAAAPURbcWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EW1FmxSTjR3TFBsUX
otQ2ZRQ0toYz1GQ0E=
Scroll Size = 611
```

```
In [26]: count = 0
list_of_LAT_LONG_pairs = []

while(scroll_size > 0):

    for inspection in results['hits']['hits']:
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None):
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
                        list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

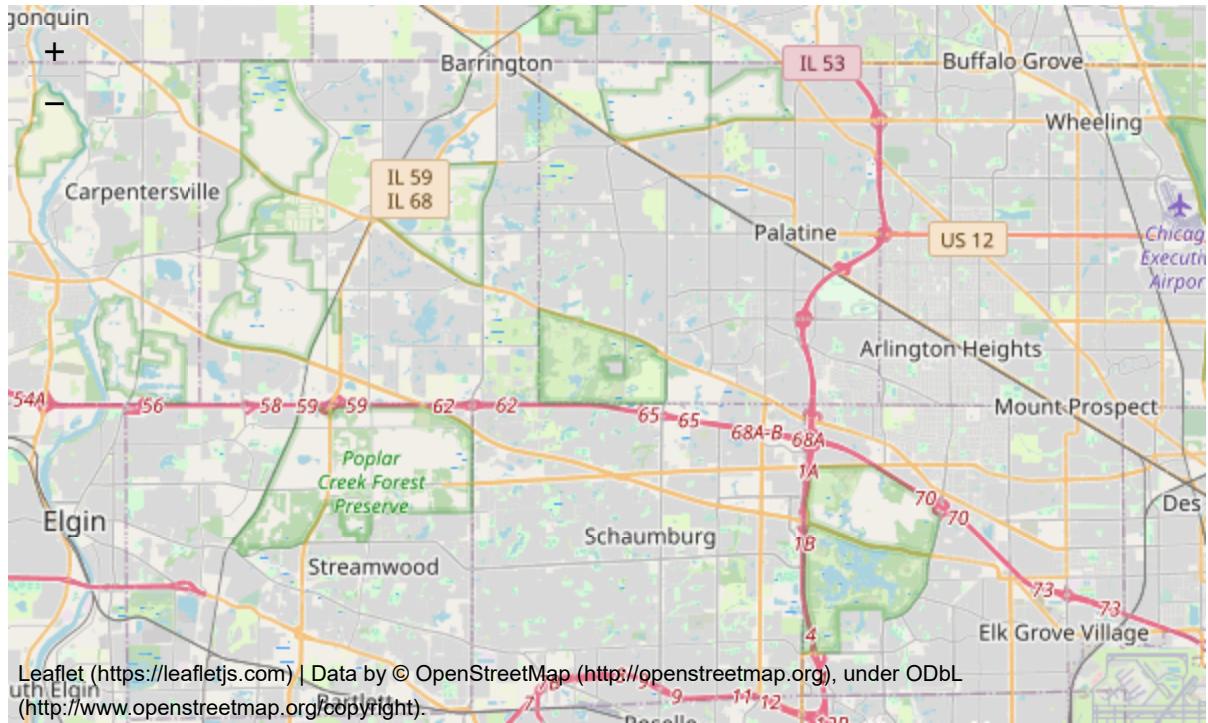
    results = es.scroll(scroll_id = sid, scroll = '2m')
    sid = results['_scroll_id']
    scroll_size = len(results['hits']['hits'])

print("Total number of match with Children's using fuzziness:",count)
```

Total number of match with Children's using fuzziness: 611

```
In [27]: chicago_map = folium.Map([41.90293279, -87.70769386], zoom_start=11)
chicago_map
```

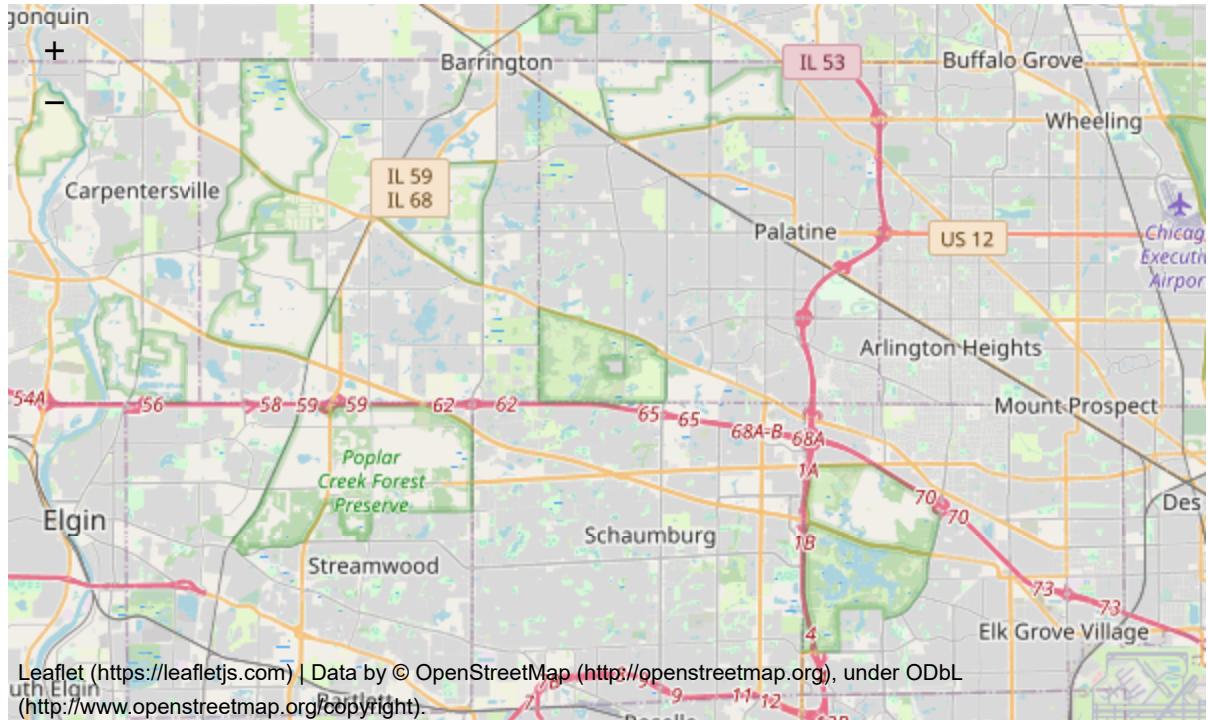
Out[27]:



In [28]: # Plot the query matches for "Children's" on Chicago HeatMap

```
chicago_map.add_child(plugins.HeatMap(list_of_LAT_LONG_pairs, radius=15))
chicago_map
```

Out[28]:



Frequent Violators

Experiment #4: Get the top list of frequent violators:

Facilities that serve children can be classified under different Facility Types:

1. Daycare Above and Under 2 Years
2. Children's Services Facility
3. Daycare (2 - 6 Years)

Use ELasticSearch and Folium to plot on the map those facilities that **failed inspection at least 5 times with risk high**.

```
In [29]: query ={
    'size' : 10000,
    'query': {
        "bool" : {
            "should": [
                {'match' : {'Facility Type': {"query" : 'Daycare (2 - 6 Years)'}}},
                {'match' : {'Facility Type': {"query" : 'Daycare Above and Under 2 Years'}}},
                {'match' : {'Facility Type': {"query" : 'CHILDRENS SERVICES'}}]
            ],
            "minimum_should_match" : 1,
            "filter" : [{"match" : {"Results": {"query": "Fail", "operator": "AND"}, "match" : {"Risk": {"query": "Risk 1 (High)", "operator": "AND"}}}]}
        ]
    },
    "aggs" : {
        "selected_dbas" :{
            "terms" : {
                "field" : "DBA Name.keyword",
                "min_doc_count": 5,
                "size" :10000
            }
        },
        "aggs": {
            "top_dba_hits": {
                "top_hits": {
                    "size": 10
                }
            }
        }
    }
}

results = es.search(index='food_inspections', body=query, scroll='1h')
```

```
In [31]: list_of_LAT_LONG_pairs = []

for dba_bucket in results["aggregations"]["selected_dbas"]["buckets"]:
    if "top_dba_hits" in dba_bucket and "hits" in dba_bucket["top_dba_hits"] and "hits" in dba_bucket["top_dba_hits"]["hits"]:

        for hit in dba_bucket["top_dba_hits"]["hits"]["hits"]:

            if "_source" in hit:

                if "Latitude" in hit["_source"] and "Longitude" in hit["_source"]:
                    list_of_LAT_LONG_pairs.append([hit["_source"]["Latitude"], hit["_source"]["Longitude"]])
```

```
# Lets dumps the LAT and LONG
# list_of_LAT_LONG_pairs
```

```
In [ ]: # Dump the hits per bucket into a dataframe object for all buckets
```

```
row_index =0
df_top_frequent_violators = pd.DataFrame()
for dba_bucket in results["aggregations"]["selected_dbas"]["buckets"]:
    if "top_dba_hits" in dba_bucket and "hits" in dba_bucket["top_dba_hits"] and "hits"
        doc_count = dba_bucket['doc_count']
        for hit in dba_bucket["top_dba_hits"]["hits"]["hits"]:
            score = hit['_score']
            if "_source" in hit:
                row_index += 1
                df_frequent_violator = pd.DataFrame(hit['_source'],index =[row_index])
                df_frequent_violator['doc_count'] = doc_count
                df_frequent_violator['score'] = score
                df_top_frequent_violators = df_top_frequent_violators.append(df_frequent_violator)
```

```
In [33]: df_top_frequent_violators
```

Out[33]:

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip
1	1319663	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0
2	1229713	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	3793.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0
3	1515476	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0
4	1229852	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	1194190.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0
5	1386187	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0
...
147	285066	THE EDS EL ALBERT AMMONS NURSER	THE EDS EL ALBERT AMMONS NURSER	15803.0	Daycare (2 - 6 Years)	Risk 1 (High)	549 E 76TH ST	CHICAGO	IL	60619.0
148	1235065	THE EDS EL ALBERT AMMONS NURSER	THE EDS EL ALBERT AMMONS NURSER	15803.0	Daycare (2 - 6 Years)	Risk 1 (High)	549 E 76TH ST	CHICAGO	IL	60619.0
149	1158446	THE EDS EL ALBERT AMMONS NURSER	THE EDS EL ALBERT AMMONS NURSER	15803.0	Daycare (2 - 6 Years)	Risk 1 (High)	549 E 76TH ST	CHICAGO	IL	60619.0
150	545232	THE EDS EL ALBERT AMMONS NURSER	THE EDS EL ALBERT AMMONS NURSER	15803.0	Daycare (2 - 6 Years)	Risk 1 (High)	549 E 76TH ST	CHICAGO	IL	60619.0

Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip
151	1234922	THE EDSEL ALBERT AMMONS NURSER	EDSEL ALBERT AMMONS NURSER	15803.0	Daycare (2 - 6 Years)	Risk 1 (High)	549 E 76TH ST	CHICAGO IL	60619.0

THE
EDSEL
ALBERT
AMMONS
NURSER

THE
EDSEL
ALBERT
AMMONS
NURSER

151 rows x 19 columns

```
In [34]: # Print the number of violations for every DBA NAME
```

```
df_top_frequent_violators['DBA Name'].value_counts()
```

```
Out[34]:
```

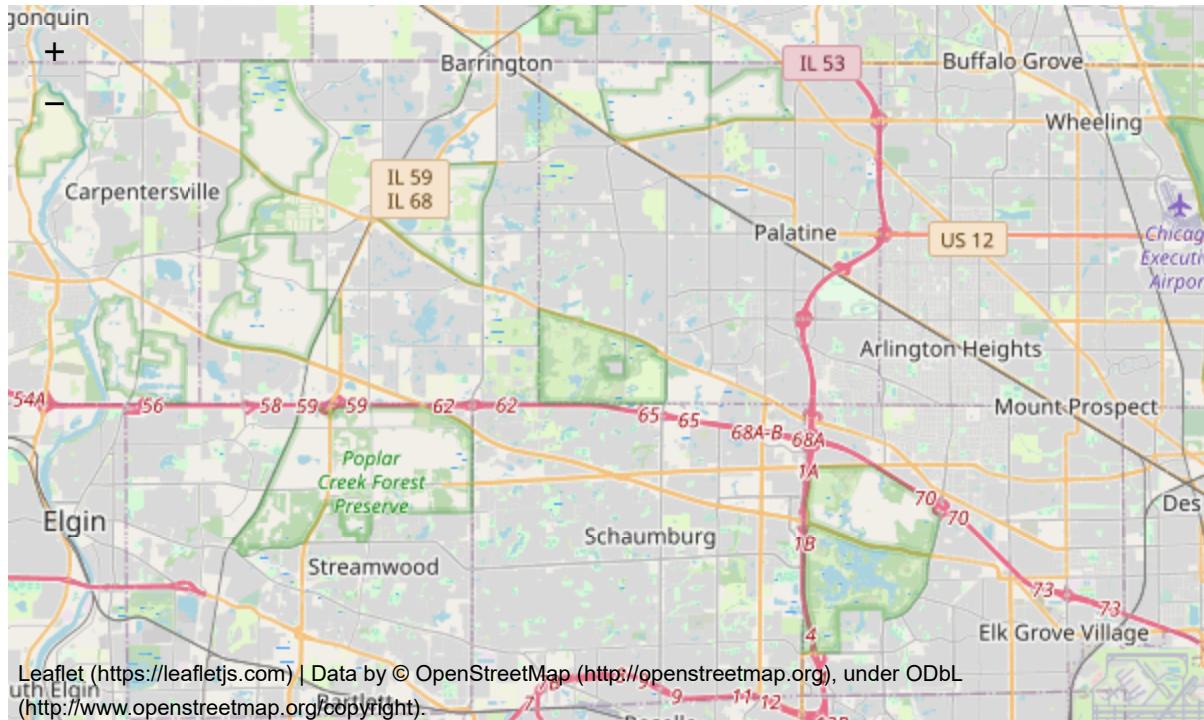
BUSY BUMBLE BEE ACADEMY DAYCARE	9
BOTTLES TO BOOKS LEARNING CENTER	8
A CHILD'S WORLD EARLY LEARNING CENTER	7
AMAZING GRACE DAYCARE CENTER	7
KIDS R FIRST LEARNING ACADEMY	6
Little People's Day Care & Kindergarten, Inc.	6
LITTLE KIDS VILLAGE LEARNING	6
LINCOLN KING DAY CARE	6
THE WORLD IS YOUR'S CHILD CARE & LEARNING CENTER INC.	6
JELLYBEAN LEARNING CENTER	6
EARLY CHILDHOOD EDUCARE CENTER	6
DISCOVERY LEARNING ACADEMY, INC.	6
COMMONWEALTH DAYCARE CENTER	6
FIRMAN COMMUNITY SERVICES	6
KENYATTA'S DAYCARE	5
THE CRYSTAL PALACE EARLY LITERACY ZONE	5
MONTESSORI ACDY. INFT/TOD. CNT	5
MOLADE' CHILD DEVELOPMENT CENTER	5
LAKE & PULASKI CHILD DEVELOPMENT CENTER	5
EZZARD CHARLES DAYCARE CENTER	5
GREATER INSTITUTE AME CHURCH	5
GRANT DAY CARE INC	5
CENTRO INFANTIL	5
ANGELS	5
ADA S MCKINLEY MAGGIE DRUMMON	5
THE EDSEL ALBERT AMMONS NURSER	5

Name: DBA Name, dtype: int64

```
In [36]: # Plot the top frequent violators on Chicago HeatMap
```

```
chicago_map.add_child(plugins.HeatMap(list_of_LAT_LONG_pairs, radius=15))
chicago_map
```

Out[36]:



In [53]: # Experiment 1 Query

```
query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{"match" : {"Results": "Fail"}}, {"match" : {"Risk": {"o
                {"query_string": {
                    "query": "*Child*", #using regex of child
                    "fields": ["Facility Type","Violations","D
                }
            }
        ]
    }
}
results = es.search(index='food_inspections', body=query, scroll='1h')
```

In [54]: # Experiment 1 Results

```
sid = results['_scroll_id']
scroll_size = results['hits']['total']
max_score = results['hits']['max_score']

print('sid = ', sid)
print('Scroll Size = ', scroll_size)
print('Max Score = ', max_score)
```

```

sid = DnF1ZXJ5VGh1bkZldGNoCgAAAAAD1EnqFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RJ7BZsUk
40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUSe4WbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EnwFmxSTjR
3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RJ6xZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUSe0WbFJONHdM
UGxRei1DZ1FDS2hjOUZDQQAAAAD1EnxFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RJ7xZsUk40d0xQb
FF6LUNmUUNLaGM5RkNBAAAAAPUSeIwbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EnxFmxSTjR3TFBsUX
otQ2ZRQ0toYz1GQ0E=
Scroll Size = 774
Max Score = 3.5609713

```

In [55]: # Experiment 1 Heatmap

```

count = 0
list_of_LAT_LONG_pairs = []
while(scroll_size > 0):

    for inspection in results['hits']['hits']:
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

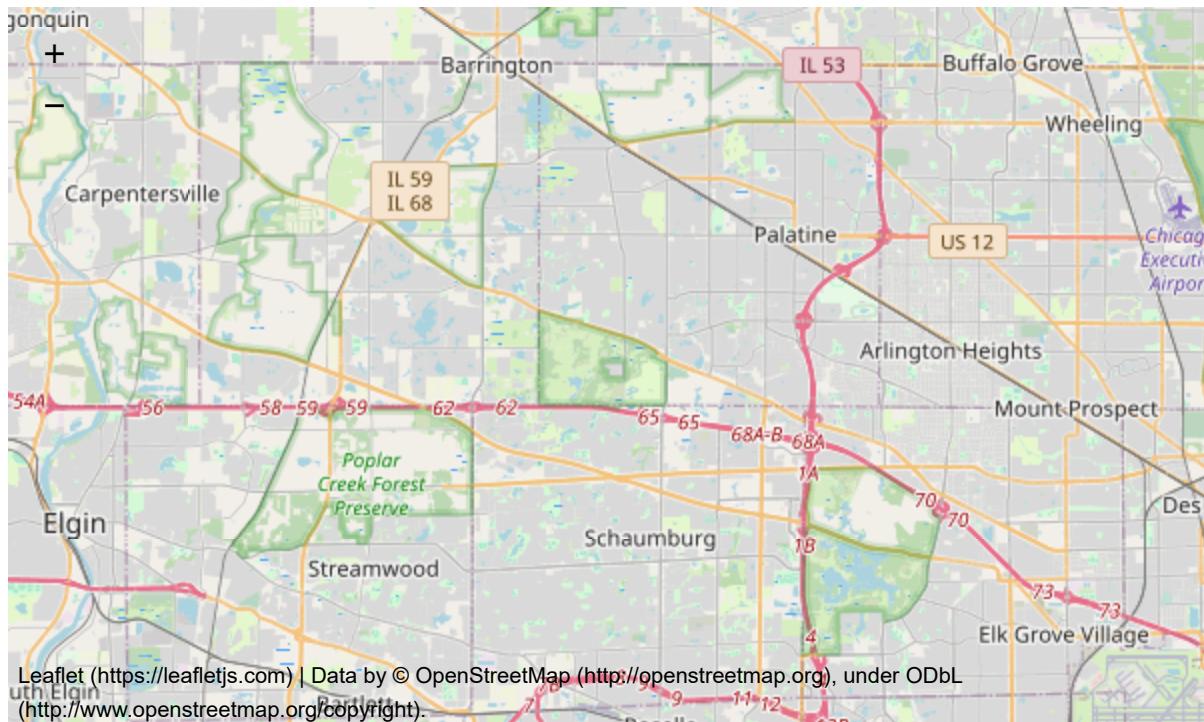
        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None):
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
                        list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

    results = es.scroll(scroll_id = sid, scroll = '2m')
    sid = results['_scroll_id']                                #Changing the scroll-id
    scroll_size = len(results['hits']['hits'])

chicago_map.add_child(plugins.HeatMap(list_of_LAT_LONG_pairs, radius=15))
chicago_map

```

Out[55]:



In [56]: # Experiment 2 Query

```
query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{ 'match' : { 'Results': 'Fail'}}], {"match" : { 'Risk' : { "c
                {"query_string": {
                    "query": "Child~2",
                    "fields": [ "Facility Type", "Violations", "D
                }
            }
        ]
    }
}
results = es.search(index='food_inspections', body=query, scroll='1h')
```

In [57]: # Experiment 2 Results

```
sid = results['_scroll_id']
scroll_size = results['hits']['total']
max_score = results['hits']['max_score']

print('sid = ', sid)
print('Scroll Size = ', scroll_size)
print('Max Score = ', max_score)
```

```
sid = DnF1ZJXJ5VGh1bkZldGNoCgAAAAAD1En0FmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RJ9hZsUk
40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUSfcWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1En4FmxSTjR
3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RJ9RzsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUSf0WbFJONHdM
UGxRei1DZ1FDS2hjOUZDQQAAAAD1En6FmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RJ-RzsUk40d0xQb
FF6LUNmUUNLaGM5RkNBAAAAAPUSfsWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1En8FmxSTjR3TFBsUX
otQ2ZRQ0toYz1GQ0E=
Scroll Size = 10970
Max Score = 7.226171
```

In [58]: # Experiment 2 Heatmap

```
count = 0
list_of_LAT_LONG_pairs = []

while(scroll_size > 0):

    for inspection in results['hits']['hits']:
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
```

```

list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

results = es.scroll(scroll_id = sid, scroll = '2m')
sid = results['_scroll_id']
scroll_size = len(results['hits']['hits'])

chicago_map.add_child(plugins.HeatMap(list_of_LAT_LONG_pairs, radius=15))
chicago_map

```

Out[58]: Make this Notebook Trusted to load map: File -> Trust Notebook

+
—

Leaflet (<https://leafletjs.com>) | Data by © OpenStreetMap (<http://openstreetmap.org>), under ODbL (<http://www.openstreetmap.org/copyright>).

In [59]: # Experiment 3 Query

```

query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{"match" : {"Results": 'Fail'}}, {"match" : {"Risk": "D"}],
                    {"query_string": {
                        "query": "Child's~2",
                        "fields": ["Facility Type", "Violations", "D"]
                    }
                }
            ]
        }
    }
}
results = es.search(index='food_inspections', body=query, scroll='1h')

```

In [60]: # Experiment 3 Results

```

sid = results['_scroll_id']
scroll_size = results['hits']['total']
max_score = results['hits']['max_score']

print('sid = ', sid)
print('Scroll Size = ', scroll_size)
print('Max Score = ', max_score)

```

```

sid = DnF1ZXJ5VGh1bkZldGNoCgAAAAAD1EoDFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RKbxZsUk
40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUSgQWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EoFFmxSTjR
3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RKCBZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUSgYWbFJONHdM
UGxRei1DZ1FDS2hjOUZDQQAAAAD1EoLFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RKCRZsUk40d0xQb
FF6LUNmUUNLaGM5RkNBAAAAAPUSg0WbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1EoMFmxSTjR3TFBsUX
otQ2ZRQ0toYz1GQ0E=
Scroll Size = 212
Max Score = 8.618152

```

In [61]: # Experiment 3 Heatmap

```

count = 0
list_of_LAT_LONG_pairs = []

while(scroll_size > 0):

    for inspection in results['hits']['hits']:
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None):
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
                        list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

    results = es.scroll(scroll_id = sid, scroll = '2m')
    sid = results['_scroll_id']
    scroll_size = len(results['hits']['hits'])

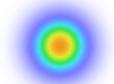
chicago_map.add_child(plugins.HeatMap(list_of_LAT_LONG_pairs, radius=15))
chicago_map

```

Out[61]: Make this Notebook Trusted to load map: File -> Trust Notebook

+

-



The results vary greatly from the previous experiment due to the fuzziness parameter staying at 2. The length of "Child" is 5, so allowing for 2 edits on the second experiment is hardly narrowing at all.

In [62]: `df_top_frequent_violators.head()`

Out[62]:

		Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip	I
1	1319663	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0	0
2	1229713	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	3793.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0	0
3	1515476	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0	1
4	1229852	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	1194190.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0	0
5	1386187	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0	0

In [63]: `# column from Experiment 4, frequency of 'DBA Name'`

```
dba_freq = pd.DataFrame(df_top_frequent_violators['DBA Name'].value_counts().reset_index())
print(dba_freq)
```

		index num_violations
0	BUSY BUMBLE BEE ACADEMY DAYCARE	9
1	BOTTLES TO BOOKS LEARNING CENTER	8
2	A CHILD'S WORLD EARLY LEARNING CENTER	7
3	AMAZING GRACE DAYCARE CENTER	7
4	KIDS R FIRST LEARNING ACADEMY	6
5	Little People's Day Care & Kindergarten, Inc.	6
6	LITTLE KIDS VILLAGE LEARNING	6
7	LINCOLN KING DAY CARE	6
8	THE WORLD IS YOUR'S CHILD CARE & LEARNING CENT...	6
9	JELLYBEAN LEARNING CENTER	6
10	EARLY CHILDHOOD EDUCARE CENTER	6
11	DISCOVERY LEARNING ACADEMY, INC.	6
12	COMMONWEALTH DAYCARE CENTER	6
13	FIRMAN COMMUNITY SERVICES	6
14	KENYATTA'S DAYCARE	5
15	THE CRYSTAL PALACE EARLY LITERACY ZONE	5
16	MONTESSORI ACDY. INFT/TOD. CNT	5
17	MOLADE' CHILD DEVELOPMENT CENTER	5
18	LAKE & PULASKI CHILD DEVELOPMENT CENTER	5
19	EZZARD CHARLES DAYCARE CENTER	5
20	GREATER INSTITUTE AME CHURCH	5
21	GRANT DAY CARE INC	5
22	CENTRO INFANTIL	5
23	ANGELS	5
24	ADA S MCKINLEY MAGGIE DRUMMON	5
25	THE EDSEL ALBERT AMMONS NURSER	5

In [64]: # Count # of Licences by 'DBA Name'

```
a = pd.DataFrame(df_top_frequent_violators[['DBA Name', 'License #']].value_counts().reset_index(name='num_licenses'))
license_count = a['DBA Name'].value_counts().reset_index(name='num_licenses')
#print(type(license_count['counts']))
print(license_count)
```

		index	num_licenses
0	JELLYBEAN LEARNING CENTER		4
1	FIRMAN COMMUNITY SERVICES		4
2	EARLY CHILDHOOD EDUCARE CENTER		3
3	MONTESSORI ACDY. INFT/TOD. CNT		3
4	CENTRO INFANTIL		3
5	KENYATTA'S DAYCARE		3
6	BUSY BUMBLE BEE ACADEMY DAYCARE		3
7	Little People's Day Care & Kindergarten, Inc.		2
8	DISCOVERY LEARNING ACADEMY, INC.		2
9	ANGELS		2
10	LAKE & PULASKI CHILD DEVELOPMENT CENTER		2
11	LITTLE KIDS VILLAGE LEARNING		2
12	GREATER INSTITUTE AME CHURCH		2
13	KIDS R FIRST LEARNING ACADEMY		2
14	AMAZING GRACE DAYCARE CENTER		2
15	THE CRYSTAL PALACE EARLY LITERACY ZONE		2
16	ADA S MCKINLEY MAGGIE DRUMMON		2
17	BOTTLES TO BOOKS LEARNING CENTER		2
18	EZZARD CHARLES DAYCARE CENTER		2
19	A CHILD'S WORLD EARLY LEARNING CENTER		2
20	THE WORLD IS YOUR'S CHILD CARE & LEARNING CENT...		2
21	LINCOLN KING DAY CARE		1
22	MOLADE' CHILD DEVELOPMENT CENTER		1
23	THE EDSEL ALBERT AMMONS NURSER		1
24	GRANT DAY CARE INC		1
25	COMMONWEALTH DAYCARE CENTER		1

```
In [65]: dbasename_violations_licenses = pd.merge(dba_freq, license_count, on='index')
print(dbasename_violations_licenses)
```

		index	num_violations	\
0	BUSY BUMBLE BEE ACADEMY DAYCARE		9	
1	BOTTLES TO BOOKS LEARNING CENTER		8	
2	A CHILD'S WORLD EARLY LEARNING CENTER		7	
3	AMAZING GRACE DAYCARE CENTER		7	
4	KIDS R FIRST LEARNING ACADEMY		6	
5	Little People's Day Care & Kindergarten, Inc.		6	
6	LITTLE KIDS VILLAGE LEARNING		6	
7	LINCOLN KING DAY CARE		6	
8	THE WORLD IS YOUR'S CHILD CARE & LEARNING CENT...		6	
9	JELLYBEAN LEARNING CENTER		6	
10	EARLY CHILDHOOD EDUCARE CENTER		6	
11	DISCOVERY LEARNING ACADEMY, INC.		6	
12	COMMONWEALTH DAYCARE CENTER		6	
13	FIRMAN COMMUNITY SERVICES		6	
14	KENYATTA'S DAYCARE		5	
15	THE CRYSTAL PALACE EARLY LITERACY ZONE		5	
16	MONTESSORI ACDY. INFT/TOD. CNT		5	
17	MOLADE' CHILD DEVELOPMENT CENTER		5	
18	LAKE & PULASKI CHILD DEVELOPMENT CENTER		5	
19	EZZARD CHARLES DAYCARE CENTER		5	
20	GREATER INSTITUTE AME CHURCH		5	
21	GRANT DAY CARE INC		5	
22	CENTRO INFANTIL		5	
23	ANGELS		5	
24	ADA S MCKINLEY MAGGIE DRUMMON		5	
25	THE EDSEL ALBERT AMMONS NURSER		5	

	num_licenses
0	3
1	2
2	2
3	2
4	2
5	2
6	2
7	1
8	2
9	4
10	3
11	2
12	1
13	4
14	3
15	2
16	3
17	1
18	2
19	2
20	2
21	1
22	3
23	2
24	2
25	1

```
In [77]: ddbname_violations_3ormorelicenses = ddbname_violations_licenses[dbname_violations_licenses['num_violations'] > 3]
print(dbname_violations_3ormorelicenses)
```

Chicago_Food_Inspection-Final

			index	num_violations	num_licenses
0	BUSY BUMBLE BEE ACADEMY DAYCARE			9	3
9	JELLYBEAN LEARNING CENTER			6	4
10	EARLY CHILDHOOD EDUCARE CENTER			6	3
13	FIRMAN COMMUNITY SERVICES			6	4
14	KENYATTA'S DAYCARE			5	3
16	MONTESSORI ACDY. INFT/TOD. CNT			5	3
22	CENTRO INFANTIL			5	3

In [66]: df_top_frequent_violators.head()

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip
1	1319663	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0 0
2	1229713	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	3793.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0 0
3	1515476	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0 1
4	1229852	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	1194190.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0 0
5	1386187	BUSY BUMBLE BEE ACADEMY DAYCARE	BUSY BUMBLE BEE ACADEMY DAYCARE	2215472.0	Daycare (2 - 6 Years)	Risk 1 (High)	6450 S COTTAGE GROVE AVE	CHICAGO	IL	60637.0 0

In [69]: print(df_top_frequent_violators[['Latitude', 'Longitude']])

```
    Latitude  Longitude
1  41.777092 -87.606004
2  41.777092 -87.606004
3  41.777092 -87.606004
4  41.777092 -87.606004
5  41.777092 -87.606004
..      ...
147 41.756551 -87.610690
148 41.756551 -87.610690
149 41.756551 -87.610690
150 41.756551 -87.610690
151 41.756551 -87.610690
```

[151 rows x 2 columns]

```
In [79]: heatmap_dba_3ormorelicenses = df_top_frequent_violators.merge(dbaname_violations_3ormorelicenses)
heatmap_dba_3ormorelicenses = heatmap_dba_3ormorelicenses[['DBA Name', 'Latitude', 'Longitude']]
print(heatmap_dba_3ormorelicenses)
```

	DBA Name	Latitude	Longitude
0	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
1	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
2	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
3	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
4	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
5	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
6	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
7	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
8	BUSY BUMBLE BEE ACADEMY DAYCARE	41.777092	-87.606004
9	EARLY CHILDHOOD EDUCARE CENTER	41.802458	-87.624433
10	EARLY CHILDHOOD EDUCARE CENTER	41.802458	-87.624433
11	EARLY CHILDHOOD EDUCARE CENTER	41.802458	-87.624433
12	EARLY CHILDHOOD EDUCARE CENTER	41.802458	-87.624433
13	EARLY CHILDHOOD EDUCARE CENTER	41.802458	-87.624433
14	EARLY CHILDHOOD EDUCARE CENTER	41.802458	-87.624433
15	FIRMAN COMMUNITY SERVICES	41.809097	-87.627599
16	FIRMAN COMMUNITY SERVICES	41.805367	-87.616633
17	FIRMAN COMMUNITY SERVICES	41.809097	-87.627599
18	FIRMAN COMMUNITY SERVICES	41.809097	-87.627599
19	FIRMAN COMMUNITY SERVICES	41.796235	-87.630405
20	FIRMAN COMMUNITY SERVICES	41.805367	-87.616633
21	JELLYBEAN LEARNING CENTER	41.765802	-87.616183
22	JELLYBEAN LEARNING CENTER	41.765802	-87.616183
23	JELLYBEAN LEARNING CENTER	41.765802	-87.616183
24	JELLYBEAN LEARNING CENTER	41.765802	-87.616183
25	JELLYBEAN LEARNING CENTER	41.765802	-87.616183
26	JELLYBEAN LEARNING CENTER	41.739343	-87.663008
27	CENTRO INFANTIL	41.902822	-87.695990
28	CENTRO INFANTIL	41.902822	-87.695990
29	CENTRO INFANTIL	41.902822	-87.695990
30	CENTRO INFANTIL	41.902822	-87.695990
31	CENTRO INFANTIL	41.902822	-87.695990
32	KENYATTA'S DAYCARE	41.759085	-87.567448
33	KENYATTA'S DAYCARE	41.759085	-87.567448
34	KENYATTA'S DAYCARE	41.759085	-87.567448
35	KENYATTA'S DAYCARE	41.759085	-87.567448
36	KENYATTA'S DAYCARE	41.759085	-87.567448
37	MONTESSORI ACDY. INFT/TOD. CNT	41.707740	-87.643003
38	MONTESSORI ACDY. INFT/TOD. CNT	41.707740	-87.643003
39	MONTESSORI ACDY. INFT/TOD. CNT	41.707740	-87.643003
40	MONTESSORI ACDY. INFT/TOD. CNT	41.707740	-87.643003
41	MONTESSORI ACDY. INFT/TOD. CNT	41.707740	-87.643003

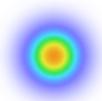
```
In [80]: heatmap_dba_3ormorelicenses_11 = list(zip(heatmap_dba_3ormorelicenses['Latitude'], heatmap_dba_3ormorelicenses['Longitude']))
```

```
In [81]: chicago_map.add_child(chicago_map.plugins.HeatMap(heatmap_dba_3ormorelicenses_11, radius=15))
chicago_map
```

Out[81]: Make this Notebook Trusted to load map: File -> Trust Notebook



Leaflet (<https://leafletjs.com>) | Data by © OpenStreetMap (<http://openstreetmap.org>), under ODbL (<http://www.openstreetmap.org/copyright>).



```
In [91]: query = {
    'size' : 10000,
    'query': {
        'bool': {
            'must' : [{ 'match' : { 'Results': 'Fail'}}, {"match" : { 'Risk' : {"o
                "query_string": {
                    "query": "((*Children*) AND (*MICE DROPPIN
                    "fields": [ "Facility Type", "Violations", "D
                }
            }
        ]
    }
}
results = es.search(index='food_inspections', body=query, scroll='1h')

sid = results['_scroll_id']
scroll_size = results['hits']['total']
max_score = results['hits']['max_score']

print('sid = ', sid)
print('Scroll Size = ', scroll_size)
print('Max Score = ', max_score)

sid =  DnF1ZXJ5VGh1bkZldGNoCgAAAAAD1HFDFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RxRhZsUk
40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUcUcWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1HFFFmxSTjR
3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RxRBZsUk40d0xQbFF6LUNmUUNLaGM5RkNBAAAAAPUcUoWbFJONHdM
UGxRei1DZ1FDS2hjOUZDQQAAAAD1HFMFmxSTjR3TFBsUXotQ2ZRQ0toYz1GQ0EAAAAAA9RxSBzsUk40d0xQb
FF6LUNmUUNLaGM5RkNBAAAAAPUcUkWbFJONHdMUGxRei1DZ1FDS2hjOUZDQQAAAAD1HFLFmxSTjR3TFBsUX
otQ2ZRQ0toYz1GQ0E=
Scroll Size =  29
Max Score =  25.657597
```

```
In [96]: count = 0
list_of_LAT_LONG_pairs = []
```

```
while(scroll_size > 0):

    for inspection in results['hits']['hits']:
        #Iterating each result
        current_location_LAT_LONG = []
        document = inspection['_source']
        count = count +1

        #defensive coding to ensure we have the fields in the inspection documents
        if 'Latitude' in document.keys():
            if 'Longitude' in document.keys():
                if 'Address' in document.keys():
                    if(document['Latitude'] != None and document['Longitude'] != None):
                        current_location_LAT_LONG.append(float(document['Latitude']))
                        current_location_LAT_LONG.append(float(document['Longitude']))
                        list_of_LAT_LONG_pairs.append(current_location_LAT_LONG)

    results = es.scroll(scroll_id = sid, scroll = '2m')
    sid = results['_scroll_id']
    scroll_size = len(results['hits']['hits'])
    #Changing the scroll-id
```