

Maryland Restaurants Inspection

Data Cleaning

```
In [1]: # import Libraries  
import numpy as np  
import pandas as pd
```

```
In [2]: # import dataset  
foodInspection = pd.read_csv (r'C:/Users/yang7/OneDrive/Desktop/python/Food_Inspection.csv')
```

```
In [3]: # get the categories  
foodInspection['Category'].unique()
```

```
Out[3]: array(['Restaurant', 'Market', 'Caterer', 'Non-Profit',  
       'Public School- Other', nan, 'Public School- Elementary',  
       'Hospital', 'Public School- Middle', 'Assisted Living',  
       'Private School', 'Nursing Home', 'Public School- High',  
       'Carry Out', 'Farmers Market', 'Snack Bar', 'Institution',  
       'Excluded Organization', 'Seasonal', 'Mobile Unit'], dtype=object)
```

After go through the names of each category, I think carry out, caterer, and restaurant are categories for restaurants.

```
In [4]: # Create the dummy variable on restaurants
for i in range(13701):
    if foodInspection.iat[i,-5]=='Carry Out' or foodInspection.iat[i,-5]=='Caterer' or foodInspection.iat[i,-5]=='Restaurant':
        foodInspection.iat[i, -5] = 1
    else:
        foodInspection.iat[i, -5] = 0
foodInspection.head()
```

Out[4]:

	Establishment ID	Name	Address 1	Address 2		City	Zip	Inspection Date	Inspection Result
0	18628	SUBWAY #22916	12193 DARNESTOWN RD.		NaN	GAITHERSBURG	20878	12/19/2018	No C Violation
1	1059	GIANT FOOD #0192	18250 FLOWER HILL WAY		NaN	GAITHERSBURG	20879	12/19/2018	C Violation Corre
2	20933	PIZZA BOLIS	7831 EASTERN AVE.		NaN	SILVER SPRING	20910	12/19/2018	No C Violation
3	22217	GREENE TURTLE	18101 TOWN CENTER DR		NaN	OLNEY	20832	12/13/2017	No C Violation
4	28408	CRAVE	7101 DEMOCRACY BLVD	SPACE #FC		BETHESDA	20817	12/13/2018	No C Violation

5 rows × 31 columns

```
In [5]: import datetime as dt
```

```
In [6]: # Create a new column for the year of the inspection.
# Create a new column for the month of the inspection.
# Create a column for the year and month.
foodInspection=foodInspection.assign(Year = ' ')
foodInspection=foodInspection.assign(Month = ' ')
foodInspection=foodInspection.assign(Month_Year= ' ')
for i in range(13701):
    foodInspection.iat[i, -3] = foodInspection.iat[i,6][-4:]
    foodInspection.iat[i, -2] = foodInspection.iat[i,6][0:2]
    foodInspection.iat[i, -1] = foodInspection.iat[i,6][0:2]+"/"+foodInspection.iat[i,6][-4:]
foodInspection.head()
```

Out[6]:

	Establishment ID	Name	Address 1	Address 2	City	Zip	Inspection Date	Inspection Result
0	18628	SUBWAY #22916	12193 DARNESTOWN RD.		NaN GAITHERSBURG	20878	12/19/2018	No C Violation
1	1059	GIANT FOOD #0192	18250 FLOWER HILL WAY		NaN GAITHERSBURG	20879	12/19/2018	C Violation Corre
2	20933	PIZZA BOLIS	7831 EASTERN AVE.		NaN SILVER SPRING	20910	12/19/2018	No C Violation
3	22217	GREENE TURTLE	18101 TOWN CENTER DR		NaN OLNEY	20832	12/13/2017	No C Violation
4	28408	CRAVE	7101 DEMOCRACY BLVD	SPACE #FC	BETHESDA	20817	12/13/2018	No C Violation

5 rows × 34 columns

```
In [7]: # Convert the Inspection_date column into a datetime column
for i in range(13701):
    foodInspection.iat[i,6]=dt.datetime.strptime(foodInspection.iat[i,6], '%m/%d/%Y')
```

```
In [8]: # For each column with the type of compliance, create a dummy variable that is 1 if the establishment is out of compliance and 0 otherwise.
for i in range(8,24):
    for j in range(13701):
        if foodInspection.iat[j,i] == 'Out of Compliance':
            foodInspection.iat[j,i] = 1
        elif foodInspection.iat[j,i] == 'In Compliance':
            foodInspection.iat[j,i] = 0
        else:
            foodInspection.iat[j,i] = np.nan
foodInspection.head()
```

Out[8]:

	Establishment ID	Name	Address 1	Address 2	City	Zip	Inspection Date	Inspection Result
0	18628	SUBWAY #22916	12193 DARNESTOWN RD.	NaN	GAITHERSBURG	20878	2018-12-19 00:00:00	No Compliance Violation
1	1059	GIANT FOOD #0192	18250 FLOWER HILL WAY	NaN	GAI	20879	2018-12-19 00:00:00	C
2	20933	PIZZA BOLIS	7831 EASTERN AVE.	NaN	SILVER SPRING	20910	2018-12-19 00:00:00	No Compliance Violation
3	22217	GREENE TURTLE	18101 TOWN CENTER DR	NaN	OLNEY	20832	2017-12-13 00:00:00	No Compliance Violation
4	28408	CRAVE	7101 DEMOCRACY BLVD	SPACE #FC	BETHESDA	20817	2018-12-13 00:00:00	No Compliance Violation

5 rows × 34 columns

In [9]: #Create a new column that contains the number of violations for that inspection (the number of categories where the establishment was not in compliance).
 foodInspection=foodInspection.assign(Number_of_Violations =0)
 for j in range(13701):
 count=0
 for i in range(8,24):
 if foodInspection.iat[j,i] == 1 or foodInspection.iat[j,i] == 0 :
 count=count+foodInspection.iat[j,i]
 foodInspection.iat[j, -1] = count
 foodInspection.head()

Out[9]:

	Establishment ID	Name	Address 1	Address 2	City	Zip	Inspection Date	Inspection Result
0	18628	SUBWAY #22916	12193 DARNESTOWN RD.	NaN	GAITHERSBURG	20878	2018-12-19 00:00:00	No Violation
1	1059	GIANT FOOD #0192	18250 FLOWER HILL WAY	NaN	GAITHERSBURG	20879	2018-12-19 00:00:00	C Violation Corrected
2	20933	PIZZA BOLIS	7831 EASTERN AVE.	NaN	SILVER SPRING	20910	2018-12-19 00:00:00	No Violation
3	22217	GREENE TURTLE	18101 TOWN CENTER DR	NaN	OLNEY	20832	2017-12-13 00:00:00	No Violation
4	28408	CRAVE	7101 DEMOCRACY BLVD	SPACE #FC	BETHESDA	20817	2018-12-13 00:00:00	No Violation

5 rows × 35 columns

```
In [10]: # Create a dummy variable that is 1 if the establishment is out of compliance
in any category.
foodInspection=foodInspection.assign(Out_Of_Compliance = 0)
for i in range(13701):
    if foodInspection.iat[i, -2] != 0:
        foodInspection.iat[i, -1] = 1
foodInspection.head()
```

Out[10]:

	Establishment ID	Name	Address 1	Address 2	City	Zip	Inspection Date	Inspe Re
0	18628	SUBWAY #22916	12193 DARNESTOWN RD.	NaN	GAITHERSBURG	20878	2018-12-19 00:00:00	No C Violation
1	1059	GIANT FOOD #0192	18250 FLOWER HILL WAY	NaN	GAI	20879	2018-12-19 00:00:00	C Violation Corre
2	20933	PIZZA BOLIS	7831 EASTERN AVE.	NaN	SILVER SPRING	20910	2018-12-19 00:00:00	No C Violation
3	22217	GREENE TURTLE	18101 TOWN CENTER DR	NaN	OLNEY	20832	2017-12-13 00:00:00	No C Violation
4	28408	CRAVE	7101 DEMOCRACY BLVD	SPACE #FC	BETHESDA	20817	2018-12-13 00:00:00	No C Violation

5 rows × 36 columns

```
In [11]: # For establishments with multiple inspections, create a new DataFrame in wide
format. Keep only the establishment ID, Category, Inspection_date, and number
of violations.
NewFoodInspection= foodInspection[['Establishment ID', 'Category', 'Inspection
Date','Number_of_Violations']].copy()
```

```
In [12]: NewFoodInspection=NewFoodInspection.sort_values('Inspection Date')
```

```
In [13]: # Reshape from long to wide (pivot) such that each establishment is a row
NewFoodInspection['idx'] = NewFoodInspection.groupby('Establishment ID').cumcount()+1
NewFoodInspection = NewFoodInspection.pivot_table(index=['Establishment ID', 'C
ategory'], columns='idx',
values=['Inspection Date', 'Number_of_Violations'],aggfunc
='first')
NewFoodInspection = NewFoodInspection.sort_index(axis=1, level=1)
NewFoodInspection.columns = [f'{x}/{y}' for x,y in NewFoodInspection.columns]
```

```
In [14]: NewFoodInspection=NewFoodInspection.assign(Num_Inspection = 0)
for i in range(len(NewFoodInspection.index)) :
    NewFoodInspection.iat[i,-1]=(34-NewFoodInspection.iloc[i].isnull().sum())/
2
NewFoodInspection=NewFoodInspection[NewFoodInspection[ 'Num_Inspection' ] > 1]
NewFoodInspection
```

Out[14]:

			Inspection Date/1	Number_Of_Violations/1	Inspection Date/2	Number_Of_Violations/2
Establishment ID	Category					
11	0	2017-11-08		2.0	2018-12-14	1.0
12	0	2018-01-23		0.0	2018-07-13	1.0
21	0	2017-10-17		1.0	2018-08-20	1.0
26	0	2017-10-11		2.0	2018-12-18	1.0
28	0	2017-10-05		0.0	2018-02-26	0.0
29	0	2018-01-25		0.0	2018-11-01	0.0
30	0	2017-10-11		0.0	2018-04-13	0.0
37	0	2017-09-28		1.0	2018-11-16	0.0
39	0	2017-12-05		1.0	2018-06-22	0.0
49	0	2017-11-08		0.0	2018-09-19	0.0
60	1	2017-11-08		1.0	2018-05-09	0.0
63	1	2018-01-18		1.0	2019-05-30	0.0
68	0	2017-09-20		2.0	2018-01-12	3.0
70	0	2017-09-21		0.0	2018-10-02	0.0
71	1	2017-09-22		0.0	2018-02-07	0.0
83	1	2017-09-06		1.0	2018-03-14	1.0
85	0	2017-10-20		1.0	2018-02-23	1.0
91	1	2018-01-25		0.0	2018-09-28	0.0
98	1	2017-10-24		0.0	2018-05-01	0.0
99	1	2018-03-13		0.0	2018-10-30	0.0
100	0	2017-09-26		1.0	2017-09-26	0.0
101	0	2018-04-18		0.0	2018-07-24	0.0

Establishment ID	Category	Inspection Date/1	Number_of_Violations/1	Inspection Date/2	Number_of_Violations/2
110	0	2017-09-19	0.0	2018-03-29	0.0
118	0	2017-09-28	0.0	2017-11-08	2.0
122	0	2017-09-07	0.0	2018-01-10	1.0
123	0	2017-10-25	0.0	2018-02-12	0.0
125	0	2017-10-02	0.0	2018-03-20	3.0
134	0	2017-12-20	1.0	2017-12-30	0.0
135	1	2017-12-20	0.0	2018-03-29	0.0
136	0	2017-10-19	0.0	2018-03-06	0.0
...
36971	0	2019-04-23	0.0	2019-07-10	0.0
36996	1	2019-07-19	0.0	2019-07-23	0.0
37000	1	2019-06-11	0.0	2019-06-18	0.0
37032	0	2019-05-14	1.0	2019-06-18	0.0
37048	0	2019-05-22	1.0	2019-08-02	1.0
37055	1	2019-05-31	1.0	2019-06-18	0.0
37066	1	2019-06-23	1.0	2019-07-01	0.0
37075	1	2019-06-17	0.0	2019-06-25	0.0
37077	0	2019-06-20	0.0	2019-08-27	0.0
37086	0	2019-08-07	1.0	2019-08-09	0.0
37090	1	2019-06-10	0.0	2019-06-24	0.0
37092	1	2019-06-26	0.0	2019-06-26	0.0
37098	1	2019-06-20	0.0	2019-07-08	0.0

Establishment ID	Category	Inspection Date/1	Number_of_Violations/1	Inspection Date/2	Number_of_Violations/2
37109	0	2019-04-30	0.0	2019-06-11	0.0
37118	1	2019-06-26	1.0	2019-08-07	0.0
37124	1	2019-06-19	0.0	2019-07-03	1.0
37131	0	2019-05-16	0.0	2019-06-04	0.0
37136	1	2019-07-30	0.0	2019-08-02	0.0
37153	1	2019-06-28	0.0	2019-07-01	0.0
37173	1	2019-07-12	0.0	2019-07-15	0.0
37174	1	2019-07-05	0.0	2019-07-08	0.0
37179	1	2019-07-02	0.0	2019-07-10	0.0
37186	1	2019-08-02	2.0	2019-08-07	1.0
37209	1	2019-08-28	1.0	2019-08-29	0.0
37210	1	2019-07-17	0.0	2019-07-18	0.0
37217	0	2019-06-27	0.0	2019-07-02	0.0
37256	1	2019-08-22	1.0	2019-08-28	0.0
37257	0	2019-08-22	0.0	2019-08-30	1.0
37281	1	2019-07-31	0.0	2019-08-22	0.0
37286	0	2019-08-29	0.0	2019-08-30	0.0

3390 rows × 35 columns

Make sure category is consistent within ID and resolve any discrepancies if necessary (i.e. each establishment has only one category).

Summary Statistics/Grouped Data

```
In [15]: # Create a table with the number of violations by violation type.
# Sort the table from the most common to least common violations.
restaurantsFood=foodInspection[foodInspection['Category'] == 1]
violations=pd.DataFrame(restaurantsFood.sum(axis = 0, skipna = True) )
violations=violations.iloc[6:22]
violations.rename(columns={0: 'Sum'}, inplace=True)
violations.sort_values('Sum', ascending=False)
```

Out[15]:

	Sum
Cold Holding Temperature (C)	1965
Rodent and Insects	1348
Hot Holding Temperature (C)	598
Food Protected from Contamination (C)	431
Proper Hand Washing (C)	376
Trans Fat Ban	209
Cooling Time and Temperature (C)	178
No-Smoking Sign Posted	142
Reheating Time and Temperature (C)	64
Hot and Cold Running Water Provided (C)	59
Food from Approved Source (C)	38
Cooking Time and Temperature (C)	31
Proper Sewage Disposal (C)	22
Toxic Substances & Pesticides	4
III Workers Restricted (C)	2
Nutritional Labeling	0

Cold Holding Temperature is the most common violation.

```
In [16]: # Prepare the dataset for the result table
comcount=[]
colnum=[]
for i in range(len(NewFoodInspection.index)):
    n=33
    while n >0:
        if NewFoodInspection.iat[i,n]>0:
            comcount.append('Never')
            colnum.append(0)
            break;
        elif NewFoodInspection.iat[i,n]==0 and NewFoodInspection.iat[i,n-2]>0:
            :
            comcount.append((n-1)/2)
            colnum.append(((n-1)/2)+1)
            break;
        else:
            n=n-2
NewFoodInspection['Times_to_compliant'] = comcount
NewFoodInspection['Colnum'] = colnum
NewFoodInspection.head()
```

Out[16]:

			Inspection Date/1	Number_Of_Violations/1	Inspection Date/2	Number_Of_Violations/2
Establishment ID	Category					
11	0	2017-11-08		2.0	2018-12-14	1.0
12	0	2018-01-23		0.0	2018-07-13	1.0
21	0	2017-10-17		1.0	2018-08-20	1.0
26	0	2017-10-11		2.0	2018-12-18	1.0
28	0	2017-10-05		0.0	2018-02-26	0.0

5 rows × 37 columns

```
In [17]: # create a new dataframe with fixed size
compliant = pd.DataFrame(index=range(16),columns=range(17))
compliant.index = [f'{x+2} inspections' for x in compliant.index]
compliant.columns = [f'After {y} Reinspections' for y in compliant.columns]
compliant.insert(0, "Never compliant", np.nan, True)
```

```
In [18]: # Create a table where each row is the number of inspections a restaurant has had and the columns are the number of reinspections until the establishment becomes compliant.
grouped = NewFoodInspection.groupby(by=['Num_Inspection','Colnum'])
for name, group in grouped:
    compliant.iat[int(name[0]-2),int(name[1])]=int(len(group))
```

```
In [19]: grouped2 = NewFoodInspection.groupby(by=[ 'Num_Inspection' ])
compliant=compliant.assign(Total = 0)
for name, group in grouped2:
    compliant.iat[int(name)-2,-1]=len(group)
```

```
In [20]: grouped3 = NewFoodInspection.groupby(by=[ 'Colnum' ])
compliant.loc['Total']=np.nan
for name, group in grouped3:
    compliant.iat[-1,int(name)]=int(len(group))
```

```
In [21]: compliant = compliant.replace(np.nan, 0)
compliant.iat[-1,-1]=int(sum(compliant['Total']))
compliant
```

Out[21]:

	Never compliant	After 0 Reinspections	After 1 Reinspections	After 2 Reinspections	After 3 Reinspections	After Reinspections
2 inspections	217.0	511	133	0	0	1
3 inspections	239.0	233	90	124	0	1
4 inspections	297.0	250	76	98	139	1
5 inspections	225.0	78	30	35	55	1
6 inspections	126.0	16	11	9	14	1
7 inspections	51.0	8	1	4	5	1
8 inspections	28.0	4	0	0	0	1
9 inspections	11.0	1	0	1	0	1
10 inspections	2.0	0	0	0	1	1
11 inspections	1.0	0	0	0	0	1
12 inspections	0.0	0	0	0	0	1
13 inspections	0.0	0	0	0	0	1
14 inspections	1.0	0	0	0	0	1
15 inspections	0.0	0	0	0	0	1
16 inspections	1.0	0	0	0	0	1
17 inspections	1.0	0	0	0	0	1
Total	1200.0	1101	341	271	214	1

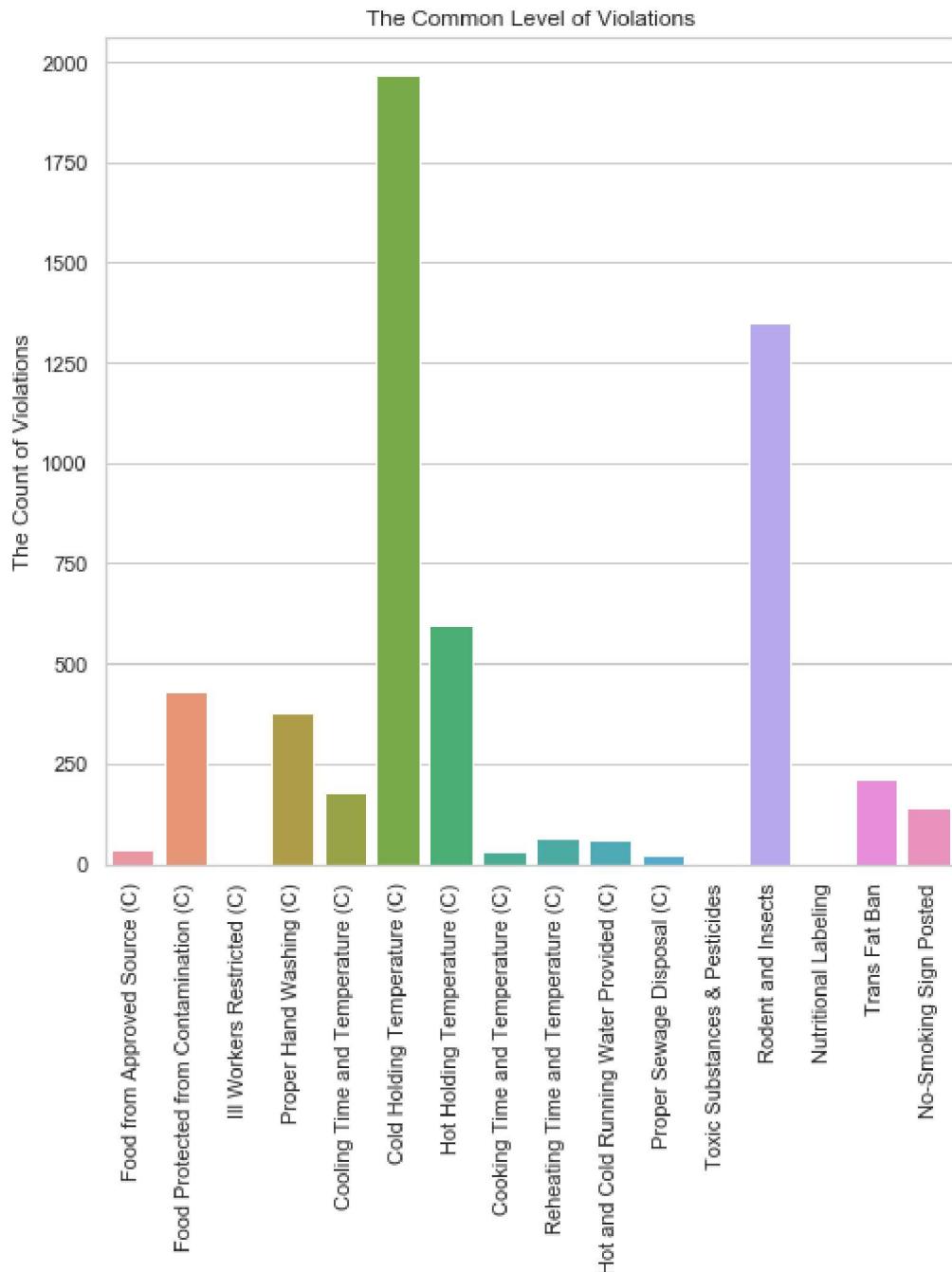
There are 3390 total restaurants haven had multiple inspections. There are 14 restaurants had over 10 inspections, and maximum is even 17. The restaurants that had more inspections tend to never compliant or be compliant after a long time. The inspection times has a positive relationship with the time that the restaurants become compliant.

Data Visualization - Mapping

```
In [22]: import matplotlib.pyplot as plt  
  
#Limit the objects to restaurants
```

```
In [23]: # Create a bar graph showing the results of 2.1
import seaborn as sns
plt.figure(figsize=(8,8))
sns.set(style="whitegrid")
sns.barplot(violations.index, 'Sum', data=violations)
plt.ylabel('The Count of Violations')
plt.title('The Common Level of Violations')
plt.xticks(rotation='vertical')
```

```
Out[23]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]),  
<a list of 16 Text xticklabel objects>)
```



```
In [24]: # Create a Line graph that shows the percent of restaurant inspections that have at least one violation by month and year.  
grouped4 = restaurantsFood.groupby(by=['Month_Year'])  
inspectionNumMonth=[]  
for name, group in grouped4:  
    inspectionNumMonth.append([name,len(group)])  
df = pd.DataFrame (inspectionNumMonth)  
df.columns = ['Month_Year', 'Num']
```

```
In [25]: restaurantsFoodViolations=restaurantsFood[restaurantsFood['Number_of_Violations'] >= 1]  
grouped5 = restaurantsFoodViolations.groupby(by=['Month_Year'])  
ViolationsNumMonth=[]  
for name, group in grouped5:  
    ViolationsNumMonth.append([name,len(group)])  
df2 = pd.DataFrame (ViolationsNumMonth)  
df2.columns = ['Month_Year', 'Num']
```

```
In [26]: # join two dfs and generate a new percentage column.
joinData=pd.merge(df, df2, on='Month_Year')
joinData=joinData.assign(Percentage = np.nan)
for i in range(len(joinData.index)) :
    joinData.iat[i,-1]=joinData.iat[i,2]/joinData.iat[i,1]
    joinData.iat[i,0]=dt.datetime.strptime(joinData.iat[i,0], '%m/%Y')
joinData=joinData.sort_values('Month_Year')
joinData
```

Out[26]:

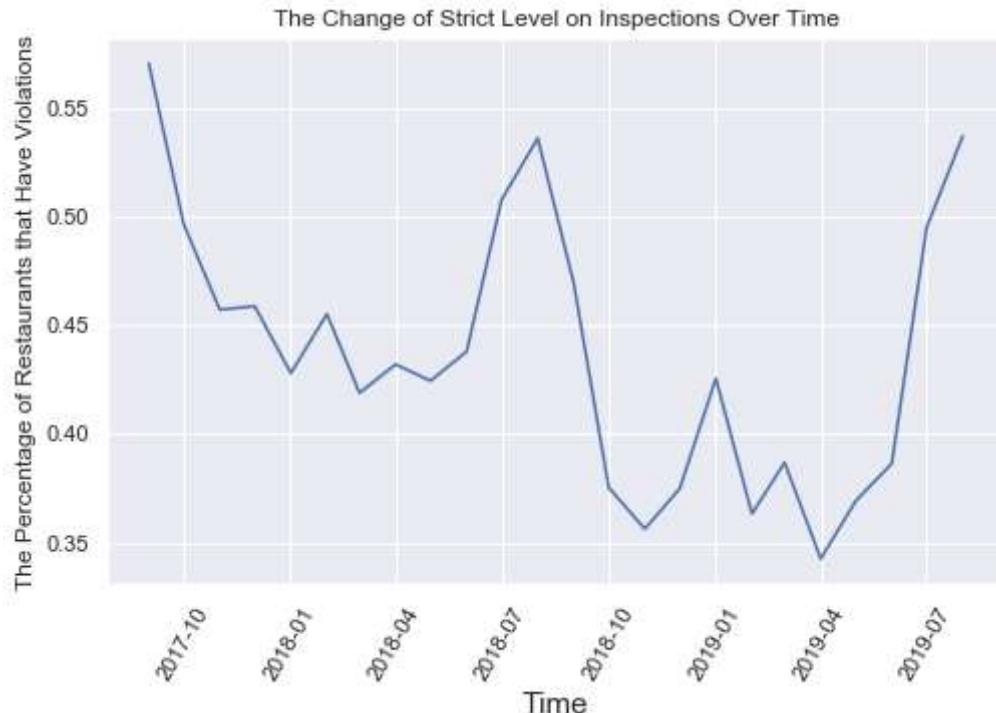
	Month_Year	Num_x	Num_y	Percentage
16	2017-09-01 00:00:00	398	227	0.570352
18	2017-10-01 00:00:00	423	210	0.496454
20	2017-11-01 00:00:00	420	192	0.457143
22	2017-12-01 00:00:00	412	189	0.458738
0	2018-01-01 00:00:00	493	211	0.427992
2	2018-02-01 00:00:00	435	198	0.455172
4	2018-03-01 00:00:00	444	186	0.418919
6	2018-04-01 00:00:00	375	162	0.432000
8	2018-05-01 00:00:00	278	118	0.424460
10	2018-06-01 00:00:00	274	120	0.437956
12	2018-07-01 00:00:00	260	132	0.507692
14	2018-08-01 00:00:00	334	179	0.535928
17	2018-09-01 00:00:00	422	198	0.469194
19	2018-10-01 00:00:00	341	128	0.375367
21	2018-11-01 00:00:00	289	103	0.356401
23	2018-12-01 00:00:00	336	126	0.375000
1	2019-01-01 00:00:00	376	160	0.425532
3	2019-02-01 00:00:00	377	137	0.363395
5	2019-03-01 00:00:00	367	142	0.386921
7	2019-04-01 00:00:00	321	110	0.342679
9	2019-05-01 00:00:00	195	72	0.369231
11	2019-06-01 00:00:00	295	114	0.386441
13	2019-07-01 00:00:00	202	100	0.495050
15	2019-08-01 00:00:00	203	109	0.536946

```
In [27]: plt.figure(figsize = (8,5))
sns.set(style="darkgrid")
line=sns.lineplot(x='Month_Year', y="Percentage",data=joinData)
for item in line.get_xticklabels():
    item.set_rotation(60)
plt.xlabel("Time", fontsize = 15)
plt.ylabel('The Percentage of Restaurants that Have Violations')
plt.title('The Change of Strict Level on Inspections Over Time')
plt.show()
```

C:\Users\yang7\Anaconda3\lib\site-packages\pandas\plotting_converter.py:129:
 FutureWarning: Using an implicitly registered datetime converter for a matplotlib plotting method. The converter was registered by pandas on import. Future versions of pandas will require you to explicitly register matplotlib converters.

To register the converters:

```
>>> from pandas.plotting import register_matplotlib_converters
>>> register_matplotlib_converters()
warnings.warn(msg, FutureWarning)
```



I do not think inspection is getting easier or harder over time, but there is certainly a pattern shown. The percentage of restaurants that have violations increases to a peak every summer. When temperature is high, the chance they get violations increases. In 2018-11 and 2019-04, most restaurants pass the inspection.

```
In [28]: import plotly.graph_objects as go
```

```
In [29]: # Your mapbox token
mapbox_access_token = 'pk.eyJ1Ijoid2FueXVuLXlhbmcilCJhIjoiY2syb3E4cTU5MTZhDNTbzNyejRxZDAzbSJ9.V9aZq1zuZ7bovxHrjfce6g'
```

```
In [30]: # Limits the points in the united states
restaurantsFoodPass=restaurantsFood[restaurantsFood['Number_Of_Violations'] == 0]
restaurantsFoodPass=restaurantsFoodPass[restaurantsFoodPass['Longitude'] >= -125]
restaurantsFoodPass=restaurantsFoodPass[restaurantsFoodPass['Longitude'] <= -70]
```

```
In [31]: restaurantsFoodViolations=restaurantsFood[restaurantsFood['Number_Of_Violations'] >= 1]
restaurantsFoodViolations=restaurantsFoodViolations[restaurantsFoodViolations['Longitude'] >= -125]
restaurantsFoodViolations=restaurantsFoodViolations[restaurantsFoodViolations['Longitude'] <= -70]
```

```
In [32]: # Add a column saying 'No violations' to the restaurants that do not have violations:
restaurantsFoodPass['Tag']='No violations'
```

```
In [33]: # Add a column with a list of violations to the restaurants that do have violations:  
restaurantsFoodViolations=restaurantsFoodViolations.assign(Violations = '')  
for j in range(3617):  
    violations=[]  
    for i in range(8,24):  
        if restaurantsFoodViolations.iat[j,i] == 1:  
            violations.append(restaurantsFoodViolations.columns[i])  
    restaurantsFoodViolations.iat[j, -1] = violations  
restaurantsFoodViolations
```

Out[33]:

	Establishment ID	Name	Address 1	Address 2	City	Zip	Inspec
4	28408	CRAVE	7101 DEMOCRACY BLVD	SPACE #FC	BETHESDA	20817	2011-00:00:00
5	28011	FIRST WATCH RESTAURANT #139	802 MUDDY BRANCH RD	NaN	GAITHERSBURG	20878	2011-00:00:00
6	27427	MIRCH MASALA GRILL	7101 DEMOCRACY BLVD	SPACE #2340	BETHESDA	20817	2011-00:00:00
7	22419	APPLEBEE'S	21048 FREDERICK RD.	NaN	GERMANTOWN	20876	2011-00:00:00
8	21231	SIRIWAN THAI RESTAURANT	736 CLOVERLY ST.	NaN	SILVER SPRING	20905	2011-00:00:00
13	19393	NATRAJ RESTAURANT CATERER	403 E. DIAMOND AVE	NaN	GAITHERSBURG	20877	2011-00:00:00
14	1264	ISKCON CHURCH	10310 OAKLYN RD.	NaN	POTOMAC	20854	2011-00:00:00
16	27538	GREGORIO'S TRATTORIA	7745 TUCKERMAN LN.	NaN	POTOMAC	20854	2011-00:00:00
19	30796	CARAVAN DELI & KABAB	615 S. FREDERICK AVE	NaN	GAITHERSBURG	20877	2011-00:00:00
20	20989	EUREST DINING SERVICES	5260 WESTERN AVE.	NaN	CHEVY CHASE	20815	2011-00:00:00
24	26642	AZUL BAR & GRILL	18749 - D N. FREDERICK RD	NaN	GAITHERSBURG	20879	2011-00:00:00
35	19043	LIN'S CHINESE RESTAURANT	734 CLOVERLY ST.	NaN	SILVER SPRING	20905	2011-00:00:00
41	2763	SUBWAY	8040 13TH ST.	NaN	SILVER SPRING	20910	2011-00:00:00

	Establishment ID	Name	Address 1	Address 2		City	Zip	Inspection Date
47	29568	LAHINCH TAVERN & GRILL	7747 TUCKERMAN LN		NaN	POTOMAC	20854	2018-03-01
53	22615	GRILLMARX STEAKHOUSE & RAW BAR	18149 TOWN CENTER DR.		NaN	OLNEY	20832	2018-03-01
55	28848	PENANG MALAYSIAN CUISINE	4933 BETHESDA AVE		NaN	BETHESDA	20814	2018-03-01
62	19256	CAPITOL BURGER	4827 FAIRMONT AVE		NaN	BETHESDA	20814	2018-03-01
72	894	BOWLMOR ROCKVILLE	15720 SHADY GROVE RD.		NaN	GAITHERSBURG	20877	2018-03-01
73	23607	MICKY'S	18204 CONTOUR RD		NaN	MONTGOMERY VILLAGE	20886	2018-03-01
74	28380	CHARLEY'S PHILLY STEAKS	7101 DEMOCRACY BLVD	SPACE FC 11		BETHESDA	20817	2018-03-01
91	29292	SAMOVAR	201 N WASHINGTON ST		NaN	ROCKVILLE	20850	2018-03-01
94	28574	PAUL BETHESDA	4760 BETHESDA AVE		NaN	BETHESDA	20814	2018-03-01
96	18993	PHO 81	19735 FREDERICK RD.		NaN	GERMANTOWN	20876	2018-03-01
109	23829	UPTOWN PIEDMONT, LLC	6720-A ROCKLEDGE DRIVE		NaN	BETHESDA	20817	2018-03-01
113	20450	BUFFALO WILD WINGS	33-C MARYLAND AVE		NaN	ROCKVILLE	20850	2018-03-01
114	11620	RUBY TUESDAY #3644	701 RUSSELL AVE.	D223		GAITHERSBURG	20877	2018-03-01

Establishment ID	Name	Address 1	Address 2	City	Zip	Inspection Date
116	251 BETHESDA BAGELS	4819 BETHESDA AVE.	NaN	BETHESDA	20814	2018-00-00
117	11362 MCDONALD'S #5226	666 QUINCE ORCHARD RD.	NaN	GAIITHERSBURG	20878	2018-00-00
120	27927 PIZZA BOLI'S	13312 OLD COLUMBIA PK	NaN	SILVER SPRING	20904	2017-10-00
122	28165 &PIZZA	7614 OLD GEORGETOWN RD	NaN	BETHESDA	20817	2018-00-00
...	2018-00-00
13602	30321 TJ CAFE	6701 DEMOCRACY BLVD, STE 107	NaN	BETHESDA	20817	2018-00-00
13604	23915 BAR LOUIE RESTAURANT	150 GIBBS ST	NaN	ROCKVILLE	20850	2018-00-00
13605	28281 EL ELCANTO RESTAURANT	8035 SNOUFFER SCHOOL RD	STE I	GAIITHERSBURG	20899	2018-00-00
13613	30245 TAKOMA BEVERAGE COMPANY	6915 LAUREL AVE	NaN	TAKOMA PARK	20912	2018-00-00
13614	18897 BLUE PEARL BUFFET & GRILL	8661 COLESVILLE RD.	NaN	SILVER SPRING	20910	2018-00-00
13617	2610 WARREN STREET DELI	9226 WARREN ST.	NaN	SILVER SPRING	20910	2018-00-00
13623	26876 REPUBLIC RESTAURANT	6939 LAUREL AVE	NaN	TAKOMA PARK	20912	2018-00-00
13626	33388 ST. VEG	14929 SHADY GROVE RD UNIT M	NaN	ROCKVILLE	20850	2018-00-00
13627	29836 TRATTORIA DA LINA	7000 CARROLL AVE	NaN	TAKOMA PARK	20912	2018-00-00

Establishment ID	Name	Address 1	Address 2	City	Zip	Inspection Date
13628	28495 RED LOBSTER #0200	15700 SHADY GROVE RD		NaN GAITHERSBURG	20877	2018-03-06
13630	19505 KABOB N KARAHII	15521 NEW HAMPSHIRE AVE		NaN SILVER SPRING	20905	2018-03-06
13632	22177 CAFE DELUXE	4910 ELM ST.		NaN BETHESDA	20814	2018-03-06
13633	18941 LIA'S RESTAURANT	4435 WILLARD AVE.		NaN CHEVY CHASE	20815	2018-03-06
13638	17838 MEIWAH RESTAURANT	4457 WILLARD AVE.		NaN CHEVY CHASE	20815	2018-03-06
13639	1214 HUNAN KITCHEN	5253 RIVER RD.		NaN BETHESDA	20816	2018-03-06
13640	17765 7-ELEVEN #11567	17701 MUNCASTER RD.		NaN DERWOOD	20855	2018-03-06
13641	22732 TANDOORI NIGHTS	7236 WOODMONT AVE.		NaN BETHESDA	20814	2018-03-06
13647	29530 NOAA GOURMET DELI	1305 EAST WESY HWY		NaN SILVER SPRING	20910	2018-03-06
13651	1345 KANPAI SUSHI	7307 MAC ARTHUR BLVD.		NaN BETHESDA	20812	2018-03-06
13654	30077 NEW VILLAGE	11540-11542 MIDDLEBROOK RD		NaN GERMANTOWN	20876	2018-03-06
13657	1800 OLD ANGLER'S INN	10801 MAC ARTHUR BLVD.		NaN POTOMAC	20854	2018-03-06
13664	1800 OLD ANGLER'S INN	10801 MAC ARTHUR BLVD.		NaN POTOMAC	20854	2017-12-06
13668	30024 FUDDRUCKERS SILVER SPRING	819 ELLSWORTH DR		NaN SILVER SPRING	20910	2017-12-06

	Establishment ID	Name	Address 1	Address 2		City	Zip	Inspec
13670	30062	ALFREDHOUSE SYMPHONY	6020 NEEDWOOD RD		NaN	DERWOOD	20855	2018-00:00
13672	18849	PRALINE BAKERY	4611-O SANGAMORE RD.		NaN	BETHESDA	20816	2018-00:00
13675	30077	NEW VILLAGE	11540-11542 MIDDLEBROOK RD		NaN	GERMANTOWN	20876	2018-00:00
13676	19498	DON POLLO OF BETHESDA	7007-7009 WISCONSIN AVE.		NaN	CHEVY CHASE	20815	2018-00:00
13679	27355	POLLERIA 3 AMIGOS II	8736 PINEY BRANCH RD		NaN	SILVER SPRING	20901	2018-00:00
13693	30024	FUDDRUCKERS SILVER SPRING	819 ELLSWORTH DR		NaN	SILVER SPRING	20910	2018-00:00
13699	30077	NEW VILLAGE	11540-11542 MIDDLEBROOK RD		NaN	GERMANTOWN	20876	2018-00:00

3617 rows × 37 columns

In [34]: restaurantsFoodPass['Longitude'].describe()

Out[34]:

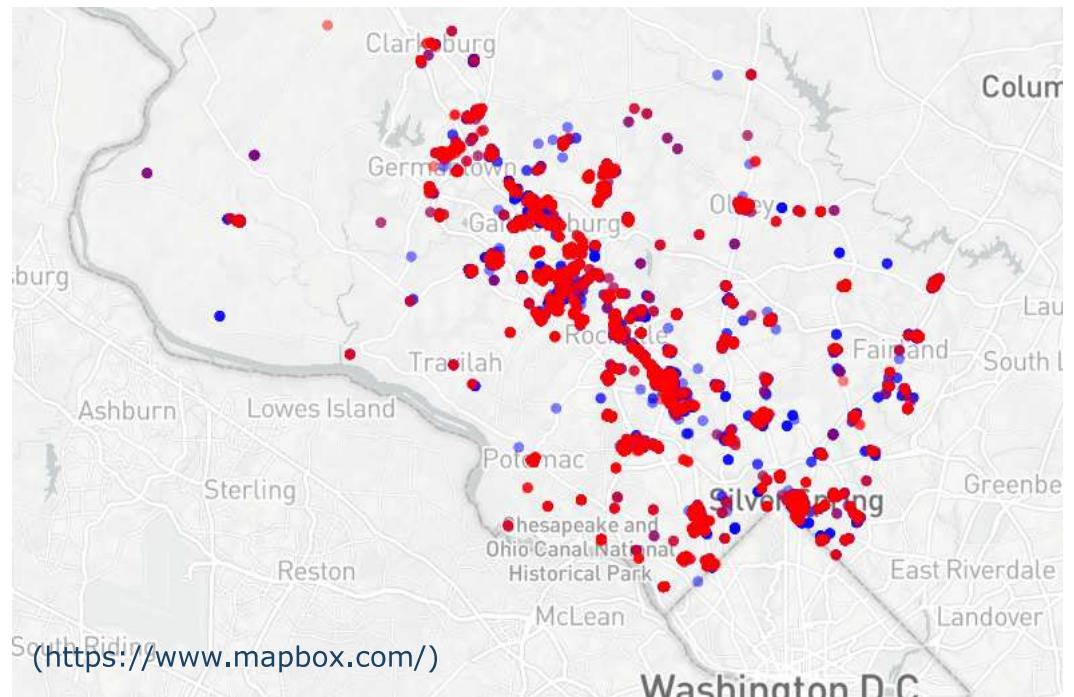
count	4632.000000
mean	-77.118721
std	0.088208
min	-77.473900
25%	-77.191925
50%	-77.116200
75%	-77.051600
max	-76.931100
Name:	Longitude, dtype: float64

```
In [35]: restaurantsFoodPass['Latitude'].describe()
```

```
Out[35]: count    4632.000000
mean      39.074284
std       0.069314
min      38.949800
25%      39.020600
50%      39.067900
75%      39.124300
max      39.949800
Name: Latitude, dtype: float64
```

```
In [36]: restaurants_map_data1 = go.Scattermapbox(  
    lon = restaurantsFoodPass['Longitude'],  
    lat = restaurantsFoodPass['Latitude'],  
    text = restaurantsFoodPass['Tag'],  
    hoverinfo='text',  
    mode = 'markers',  
    marker = dict(  
        color = 'blue',  
        symbol = 'circle',  
        opacity = .5  
    ),  
    name = "Restaurants without violations"  
)  
  
restaurants_map_data2 = go.Scattermapbox(  
    lon = restaurantsFoodViolations['Longitude'],  
    lat = restaurantsFoodViolations['Latitude'],  
    text = restaurantsFoodViolations['Violations'],  
    hoverinfo='text',  
    mode = 'markers',  
    marker = dict(  
        color = 'red',  
        symbol = 'circle',  
        opacity = .5  
    ),  
    name = "Restaurants with violations"  
)  
  
restaurants_map_layout = go.Layout(  
    title = 'Restaurants Health Inspection',  
    mapbox=go.layout.Mapbox(  
        accesstoken=mapbox_access_token,  
        zoom=1  
    )  
)  
  
restaurants_map = go.Figure(data=[restaurants_map_data1, restaurants_map_data2], layout=restaurants_map_layout)  
restaurants_map.update_layout(  
    hovermode='closest',  
    mapbox=go.layout.Mapbox(  
        accesstoken=mapbox_access_token,  
        bearing=0,  
        center=go.layout.mapbox.Center(  
            lat=39.074284,  
            lon= -77.118721  
        ),  
        pitch=0,  
        zoom=9  
    )  
)  
restaurants_map.show()
```

Restaurants Health Inspection



The restaurants that have violations tend to be located along the traffic lines and in cities. Most restaurants with violations in Rockville and Wheaton have cold holding temperature issue. However, in silver springs, there are more restaurants that have rodent and insect problem.

END