Code

An Analysis of the Spatial Qualities of Food Safety Violations in Philadelphia

Rashon Clark

1 Introduction

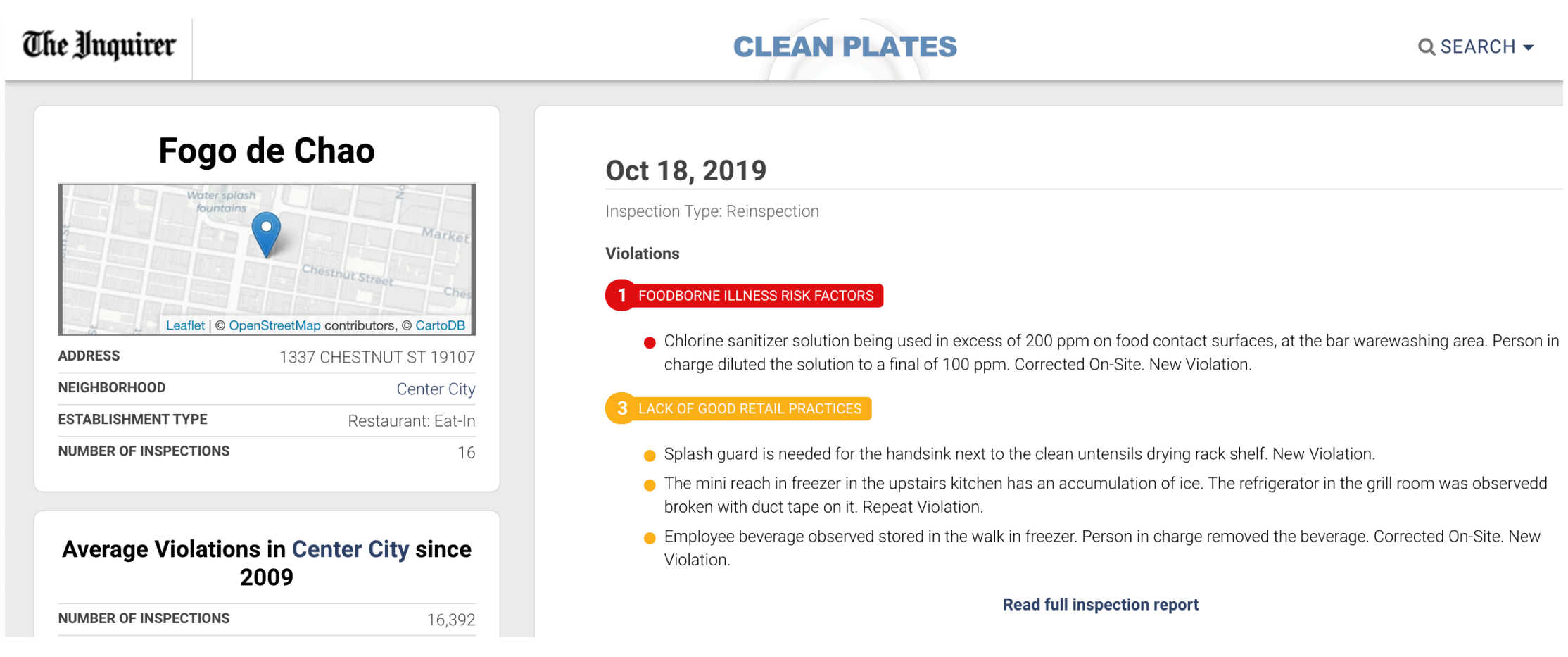
Although once mundane and bureaucratic, food safety inspections have now entered the mainstream American psyche, becoming grist for reality television, newspaper headlines, and restaurant rating systems. In a similar fashion, data on food safety has become progressively more open as large municipalities make available their immense caches of inspection reports, now often immediately digitalized. Attached to geographic markers and taken at regular intervals, these records are perfectly translatable into spatial analyses. Consequently, restaurant inspection reports offer a clear opportunity to better understand environmental influences on food safety. This is particularly relevant for food safety violations related to vermin, which are mobile and perhaps even transferrable from one property to another. This project intends to explore these qualities in the city of Philadelphia, which has an ever-growing collection of restaurant inspections, going back more than a decade.

2 Methodology

2.1 Data Collection

The project begins with the collection of data. A dataset of restaurant inspections is created by using python to web-scrap Clean Plates, a publicly available repository of Philadelphia restaurant inspections provided by the Philadelphia Inquirer and Philadelphia's Department of Health. The inspection records on this site include information on inspection type (i.e., initial inspection, complaint response), the date of the inspection, the number of food safety and poor retail practice violations, and most importantly the geographic coordinates of every restaurant, and the raw text of every violation. The raw text will be mined to create new fields to denote whether or not a restaurant featured a vermin related violation.

Clean Plates Data Repository

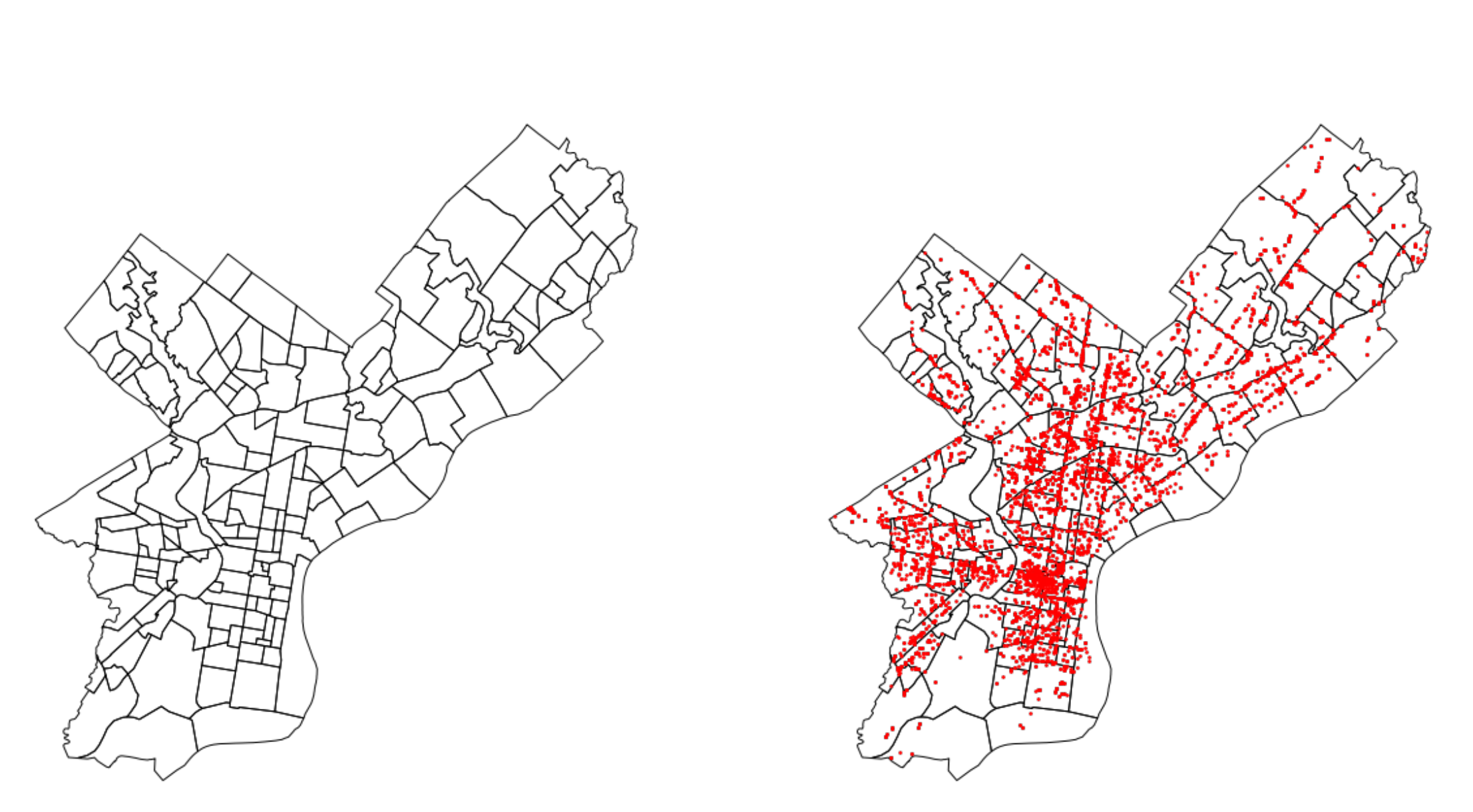


The web-scrapping endeavor is facilitated by a list of restaurants in Philadelphia compiled by Yelp and accessed through their API. A script was created that feeds these restaurant names into the search function of the violation site, and then extracts the results embedded in the HTML. The output is a Pandas data frame with relevant information on the name, date, type, coordinates, and full text of every violation. This cache of data featured about 17,000 unique establishments and nearly a million violations over a number of inspections from 2009 to the present. It is important to note that inspections extend beyond restaurants, to any institution that serves food, including schools, child-care centers, cafeterias, corner-stores, and even kiosks, although I will typically refer to them all as restaurants in this report for the sake of simplicity. It is also important to note that most restaurants and establishments in Philadelphia are inspected at least once a year, often more if they commit serious violations or have reported complaints.

Complete Dataframe of Inspection Data

	index	Name	Address	Type	Dates	Inspection_Type	Foodborne_Violations	Poor_Retail_Violations	Description	Lat	Lng
0	0	Reading Rainbow Learning Center	836 N 03RD ST 19123	Day Care: Child	Mar 4, 2013	Follow-up	1	9	Course certificate present. Serv Safe: Noemi P...	39.964001	-75.14281
1	0	Reading Rainbow Learning Center	836 N 03RD ST 19123	Day Care: Child	Mar 4, 2013	Follow-up	1	9	Provide thermometers in classroom refrigerator...	39.964001	-75.14281
2	0	Reading Rainbow Learning Center	836 N 03RD ST 19123	Day Care: Child	Mar 4, 2013	Follow-up	1	9	Minimal mouse droppings were observed next to ...	39.964001	-75.14281
3	0	Reading Rainbow Learning Center	836 N 03RD ST 19123	Day Care: Child	Mar 4, 2013	Follow-up	1	9	Several pieces of Non NSF/ANSI approved equipm...	39.964001	-75.14281
4	0	Reading Rainbow Learning Center	836 N 03RD ST 19123	Day Care: Child	Mar 4, 2013	Follow-up	1	9	Test strips for quaternary ammonium test strip...	39.964001	-75.14281

Mice Related Violation Displayed Geographically Across Philadelphia Neighborhoods



The final portion of this step is to search the violation texts for instances of vermin: roaches, rodents, and flies. I compiled a list of vermin related keywords: mouse, mice, rats, rodents, roaches, flies, vermin, and so forth, to isolate these specific types of violations. Some violations refer to the potential, rather than the actual presence of vermin, such as poor sanitation practices that invite rodents. These false positives are flagged and filtered out by locating words such as prevent, which are consistently used in this manner in inspection reports.

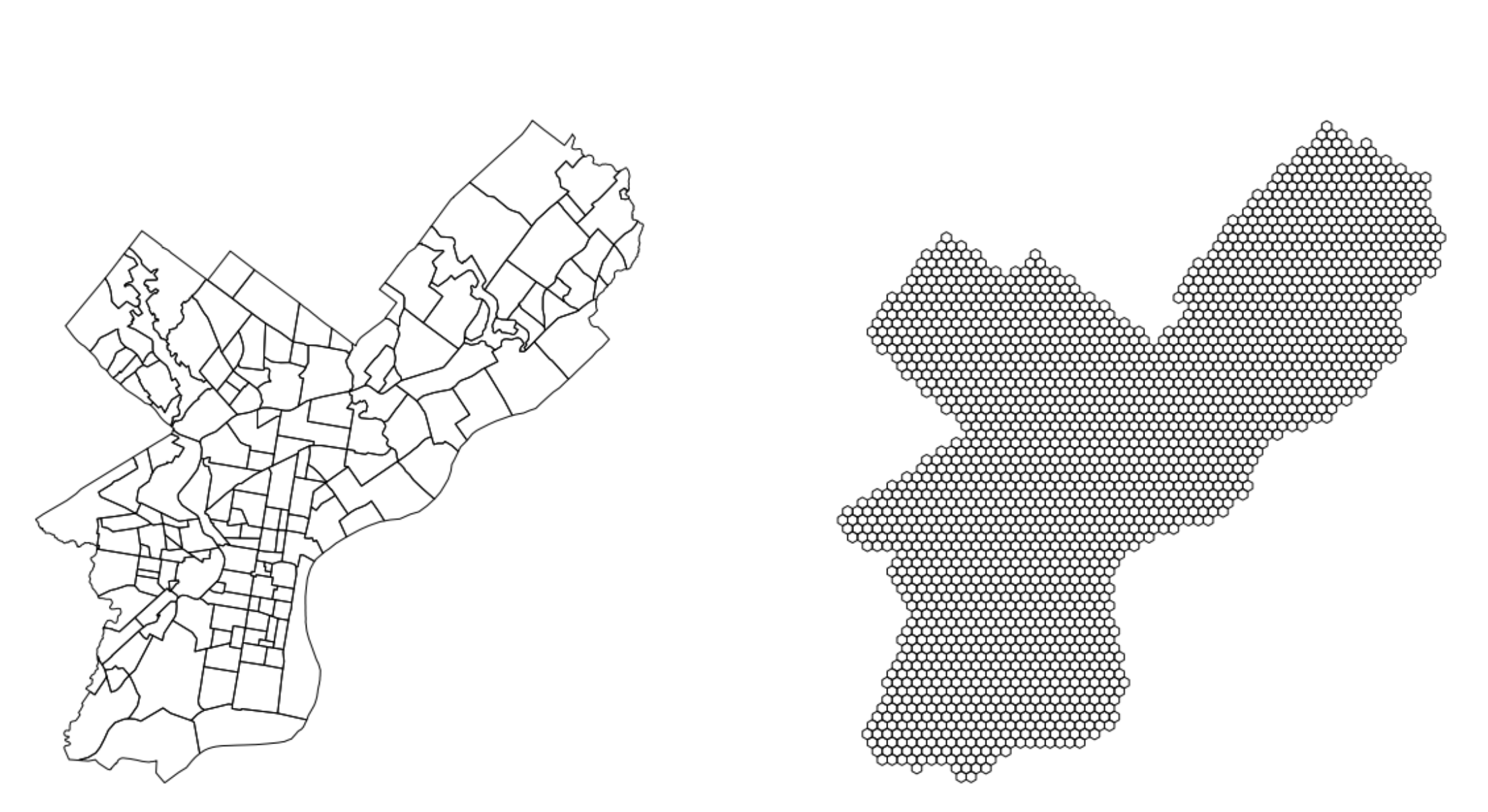
Filtered Violation Description Example

```
' Outer opening in the food facility does not protect against the entry of insects, rodents, and other animals. Pro
vide metal door sweepers under the doors to prevent vermin entry. Repeat Violation. '
```

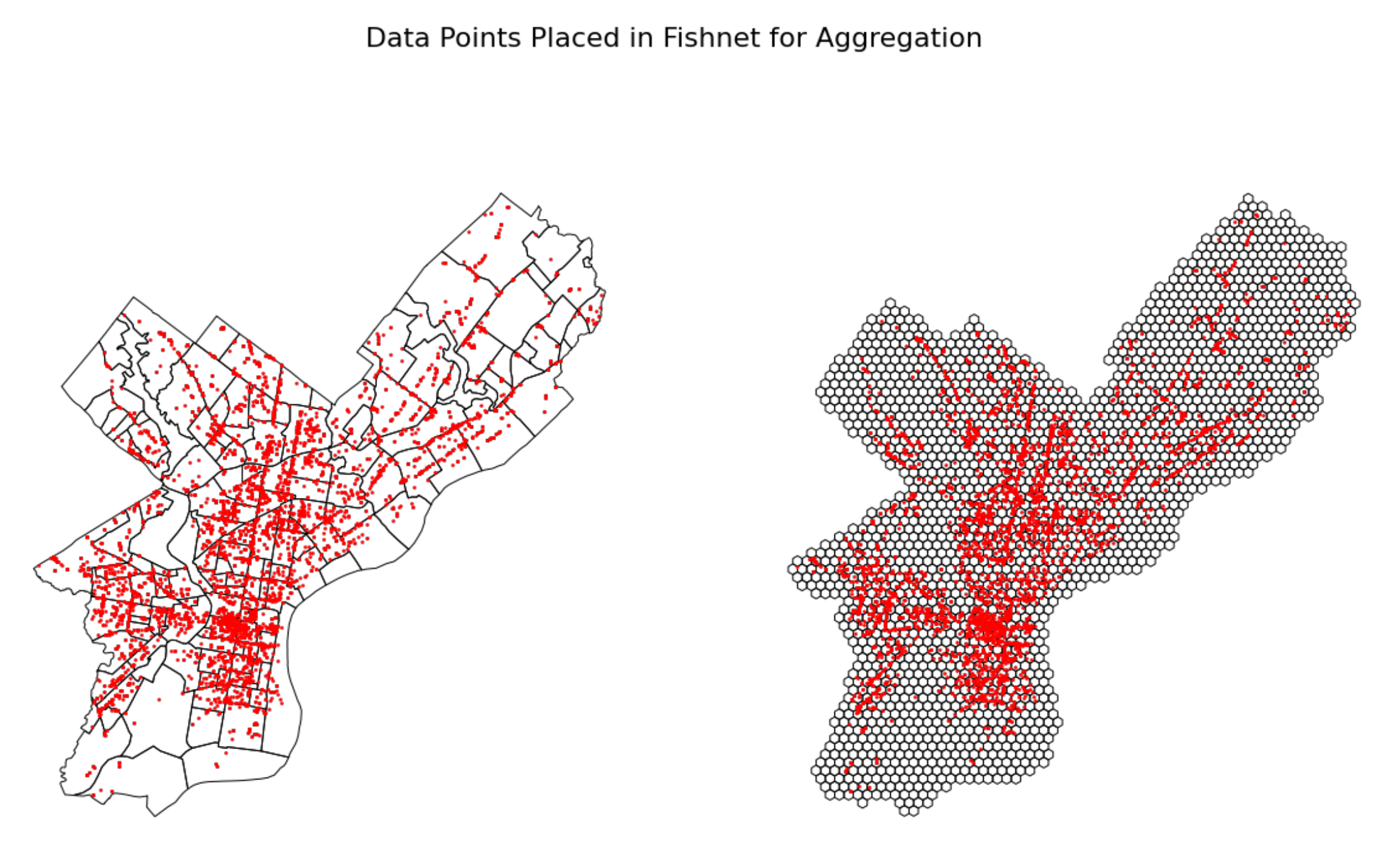
2.2 Spatial and Temporal Aggregation of Data

Now that we have our data, it is time to analyze it. We want to know if there are any vermin related spatial trends. Although a plot of violations can give us a visualize approximation of geographic clusters, we need a more defined approach for measurement. I have selected the method of fishnet, which essentially throws a fishnet of equally sized grid cells across the area of interest and counts the number of instances in each cell. For this project, I selected a cell size of 600 square meters, which is roughly the size of a city block in Philadelphia. In a way, this is a block by block analysis of Philadelphia, which is often how vermin infestation and "contagion" is publicly conceived, rightly or wrongly.

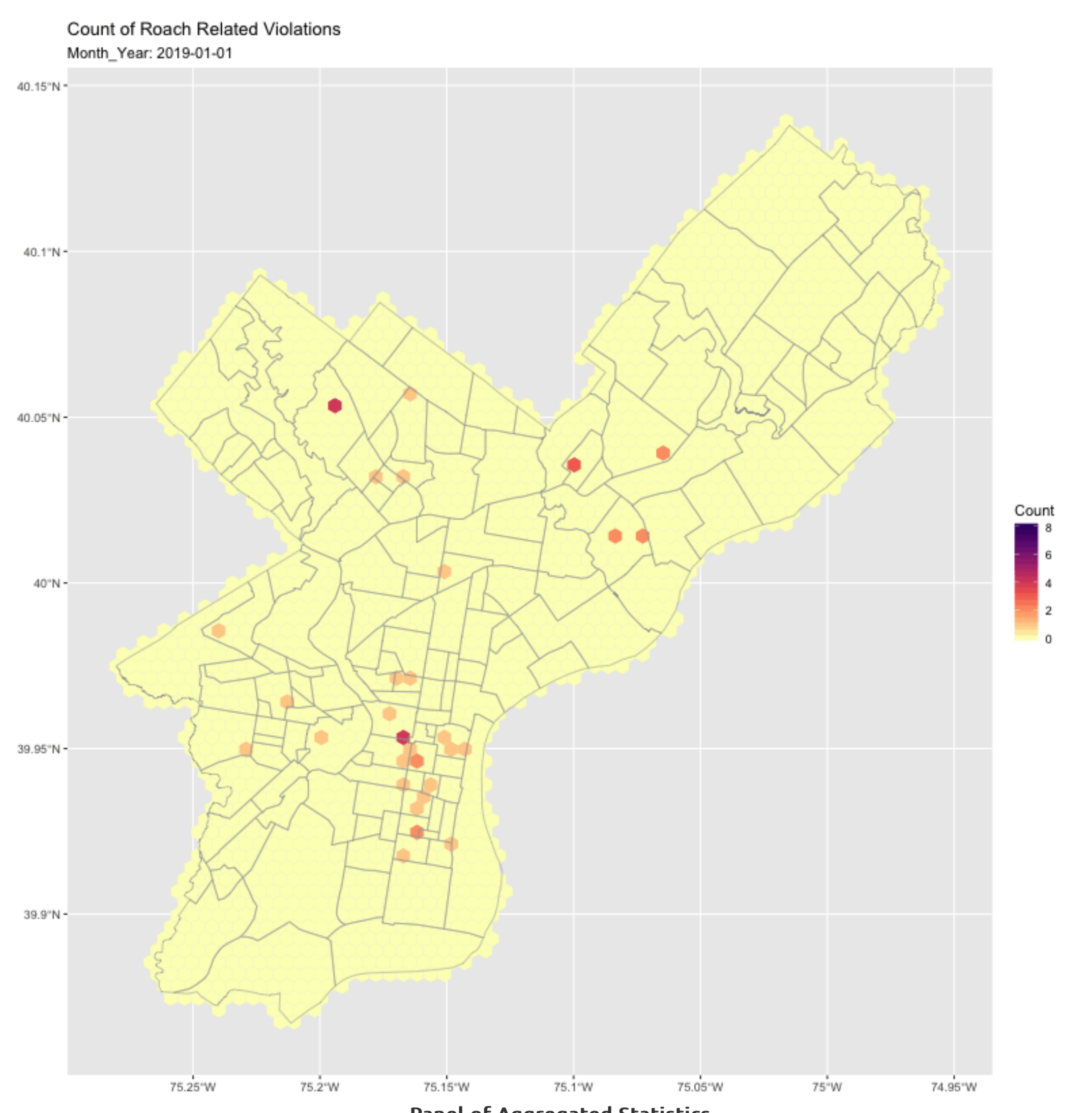
Transformation of Philadelphia into Fishnet of Grids



Data Points Placed in Fishnet for Aggregation



Each grid is given a unique ID so that they can be tracked over time. Although the majority of work in this project is completed in python, the creation of grids is performed in R, where it is much more convenient. With the date component of our data, we are also able to separate the grid cells by month and year. Consequently, we have a complete grid of Philadelphia for every month between January 2009 and February 2021. With this information, we can follow the number of vermin related violations in a specific 600 square meter area across a decade. This will allow us to locate hotspots in certain areas, or areas that are consistently beset with certain violations. In the chart below, the data is visualized through time and space. For the year of 2019, the count of roach related violations are counted for each grid for each month, which is then illustrated in an animation.



Panel of Aggregated Statistics

Legend	uniqueID	Month_Year	Fly_Count	Mice_Count	Roach_Count	Unique_Inspections	Unique_Rest_Flies	Unique_Rest_Mice	Unique_Rest_Roach	year
0	1	2009-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2009
1	1	2009-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2009
2	1	2009-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2009
3	1	2009-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2009
4	1	2009-05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2009

ut; double click to hide output

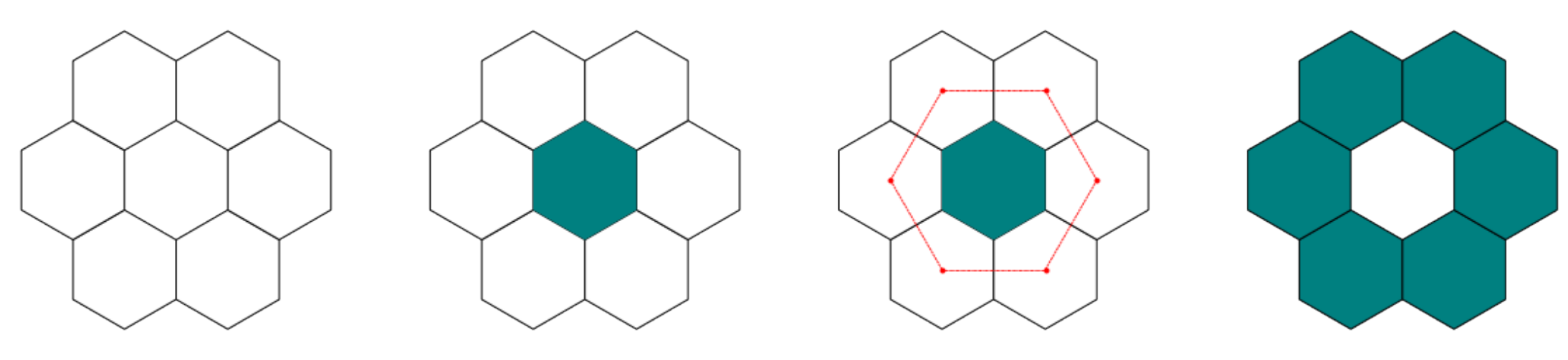
Before moving on it is important to note that the raw number of violations may be misleading, not only because there are differing numbers of restaurants across grid areas, but also because a particularly egregious restaurant can dramatically tilt the count of an area. To offer alternative metrics, I created fields which measure the ratio between the number of inspections and vermin related violations in a given cell, and the count of restaurants with at least one vermin related violation, which I distinguish as "unique restaurant violations". This latter field will lessen the bias introduced by badly performing restaurants, especially because the difference between 4 or 10 violations may not be as significant as the fact that any of them is related to vermin.

2.3 Spatial Lag

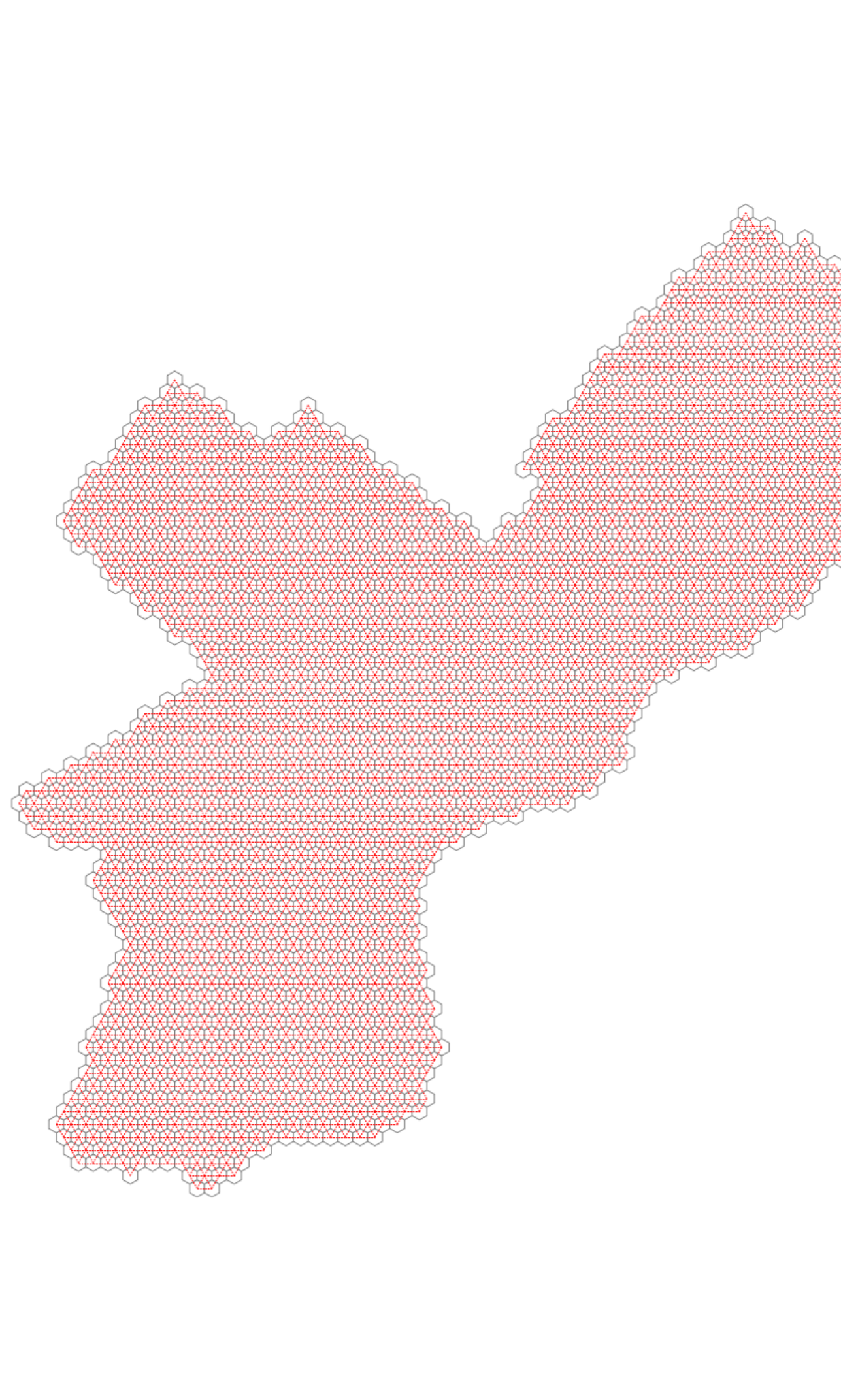
The object of this project is to answer the question of whether vermin related violations cluster in certain areas. This is known as spatial autocorrelation. We already assume that this is the case because restaurants are clustered in certain areas of the city. To answer a more finely grained question, do reports of vermin in one area anticipate reports of vermin in surrounding areas, we need additional metrics. One metric is spatial lag, which averages of a chosen statistic in surrounding area. This project employs hexagons of uniform size, meaning our spatial lag will measure the average of the surrounding six hexagons. Afterwards we can compare this average, the spatial lag, with that of the original grid cell. In more direct terms, we can compare a block and its restaurants to the restaurants of the roughly six surrounding blocks. The difference in a value such as vermin related violations can help us understand if these types of violations are solely restaurant specific (i.e., poor management practices) or connected to some type of geographic condition.

To employ this concept of spatial lag, we have to identity the "neighbors" of each grid cell. This is calculated through a python library known as Pysal. In the following plots, we can see the visualization of the neighbor relationship. Because this particular package only accounts for the relationship spatially, and not for every month, a separate script was created to calculate the neighborhood relationship across time. This produces spatial lag metrics for every grid cell, for every month. An additional step to determine the presence of a "contagion effect" would be to create a temporal lag metric that would compare values to proceeding and succeeding months, however this is unfortunately beyond the scope of this project.

Identification of Grid Cell Neighbors



Grid Neighbor Relationships Across Philadelphia



2.4 Spatial Difference

The final step in creating a useful statistic for analysis is finding the difference between the newly created spatial lag of a certain grid and its own statistics. In this way we can compare a grid to its neighbors. A low number or zero would indicate that a grid cell varies little from its neighbors, while a high number would indicate the opposite. One complication is the high presence of zeros in the data. Similar to a Poisson distribution, common for count data of this type, zeros dominate the data, lowering our newly created spatial difference statistic across the map. Nearly half of the map of Philadelphia has no restaurants, so this statistic is capturing that there is little difference between zero account grids and their neighbors. An average of spatial difference for unique restaurants with at least one vermin violation was less than 1, indicating this bias towards zeros.

```
Final_Vars_Net['Vermin_Per_Inspections_Differential'] = [Final_Vars_Net[
        'Unique_Vermin_Count_Per_Inspections'][i] - Final_Vars_Net[
        'Unique_Vermin_Count_Per_Inspections_lag'][i] for i in range(len(Final_Vars_Net))]
```

```
Final_Vars_Net['Vermin_Per_Inspections_Differential']
```

```
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
         ...
318421    0.0
318422    0.0
318423    0.0
318424    0.0
318425    0.0
Name: Vermin_Per_Inspections_Differential, Length: 318426, dtype: float64
```

```
Final_Vars_Net['Vermin_Per_Inspections_Differential'].mean()
```

```
0.0005329120598053064
```

One way to account for this is to remove all the grids that have zero counts across their entire timespan. This was not possible for this project because of the extensive computational time needed to recalculate all of the statistics. This would still not completely account for this bias towards zeros as the majority of grid cells experience zero vermin related violations on a monthly or even yearly basis.

3 Conclusion: Problems and Future Possibilities

Although the process illustrated in the previous sections demonstrate a useful way to compare spatial phenomena across time, and there are many ways to improve this project with more time or expertise. One obvious improvement could be the use of Moran's I, a method for measuring spatial autocorrelation. I avoided the use of Moran's I because the complicated manner of interpreting these statistics across many iterations through time and space was not completely transparent to me. Additionally, the computational time needed to run Moran's I tests on over 700,000 rows of data seemed computational impossible for my computer. Consequently, I found the related spatial lag statistic more useful. Nevertheless, it comes with its own limitations. Traditionally used in prediction models, it was more difficult for me to apply it innovative manner.

As noted earlier, a second problem was the dominance of zeros in the data. This is to be expected because of the nature of count data, however, similar to spatial lag, this is usually accounted for in different types of regressions and prediction models. For example, some packages allow for the use of a zero inflated binomial regression to account for zeros in data, however my grasp of this type of math is not deep enough to fully apply related techniques to my own project.

Lastly, this project does not fully account for temporal differences in its method. The use of a temporal lag, combined with spatial lag could improve the overall usefulness of a final statistic, however this was avoided due to time constraints. Overall, the visualization of the spatial aspects of food inspections would be very illustrative to a lay audience, but more work is needed to extract insights for those in urban planning or public health professions.