Mini Project 3 - Restaurant Inspection Data Analysis

Sarah Adilijiang, Yanqing Gui, Hanyang Peng

Question 1: Geographical clusters

Problem 1

1. Data Preprocessing

First, we will double check whether there is any missing value in the data set. The results show that, after the previous data cleaning in the given code, there are no more missing values left in the data set. And there are 3025 rows in total in the data1. Note that this data website updates the published data every week, thus when we download the data, it is the version of March 5th. Therefore, the number of rows is not the same as indicated in the homework.

Then, for the convenience of our data processing, we add the columns of date and day-of-week in the data set. The date column is the date of the restaurant inspection, in form of “year-month-day”. And the day-of-week column is the day of week information of the corresponding date. For example, 2017-09-28 is Thursday. From R, we know that the date of the recorded inspections is from 2016-10-04 to 2019-10-03.

What’s more, as it is mentioned in the assignment, there are several risk categories: None, Low Risk, Moderate Risk, and High Risk. As these categories are ordinal and the higher the risk the worse the general condition of the restaurant is. So we can convert the risk categories into ordinal numbers. We set risk “None” equals 1 and risk “High Risk” equals 4. In this way, we can perform further comparing or suming of the risks for each restaurant.

Finally, by viewing the data set, we can see that if a restaurant has several risk violations on a single date, then it will show multiple risk entries of a restaurant on the same date. To deal with this problem, we aggregate the data by (business_id, date) and create another data set. For each row of (business_id, date), we calculate the number of risks reported and the sum of the risks. Therefore, now there are only one entry for a restaurant on a specific date. After the aggregation, there are 1905 unique inspections in the new data set - data2.

2. Exploratory Data Analysis

2.1 Aggregate data by date

After the data preprocessing, now we can continue to explore our data set. First, we aggregate the data by date, and find there are 644 unique dates, among which the most amount of dates

(169 days) have only one unique inspection per day. And the largest number of inspections carried out in one day is 12 restaurants.

```
t(count(data3, id_count))
```

```
##          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## id_count    1    2    3    4    5    6    7    8    9    10    11    12
## n         169  132  134   93   57   29   14   11    1     1     2     1
```

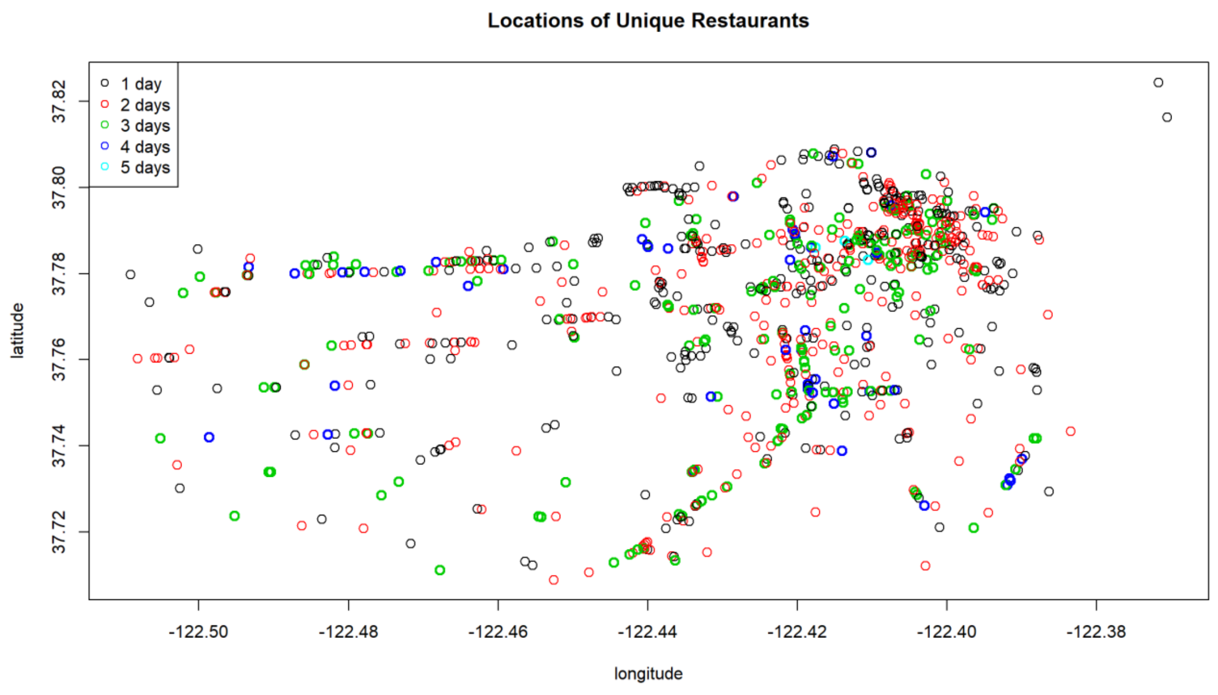
2.2 Aggregate data by business_id

Then we aggregate data by business_id, and find there are 997 unique restaurants, among which the largest proportion of the restaurants (385 restaurants) have been inspected only once in three years. The largest number of days of inspections carried out for one restaurant is 5 days in total, which only accounted for three cases.

```
ids = data4$business_id   # all unqiue business_id's
t(count(data4, date_count))
```

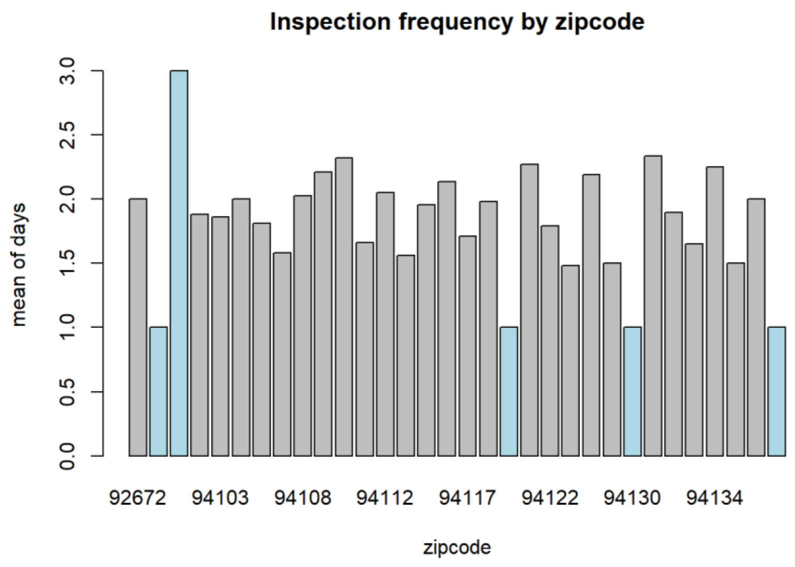
```
##            [,1] [,2] [,3] [,4] [,5]
## date_count    1    2    3    4    5
## n           385  367  197   45    3
```

2.3 Locations of unique restaurants



Then we plot the locations of unique restaurants with different colors representing different numbers of days of inspections that have been carried out for each restaurant. We can see that there are two “outlier-like” points on the top right of the map. After digging into their latitudes and longitudes, we find that they are located at the Treasure Island, which is a little away from the main San Francisco area, isolated by the seawater.

From this plot, we can see that each level of inspection frequency spreads on the map "evenly". There is no clear pattern of higher frequency of inspections in some regions shown on the map. To verify it more clearly, we draw the mean of days of inspections by the zipcode region.



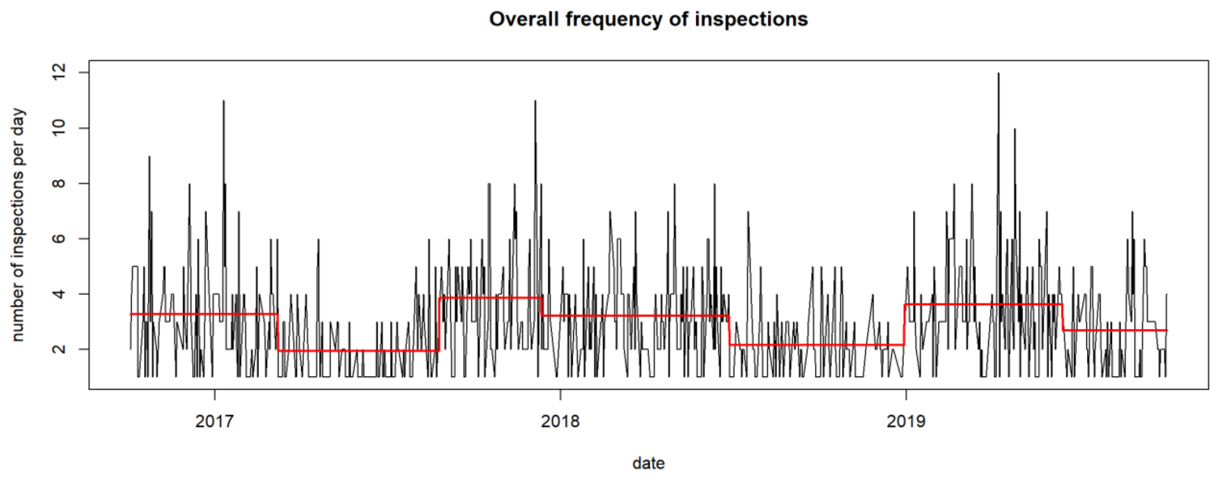
From the bar plot, we can see that most of the regions have similar inspection levels, though there are four regions that have lower levels (1 inspectioned day) and one region that has higher inspection-levels (3 inspectioned days). We extract these five zipcode regions to explore the reason.

zipcode <dbl>	id_count <int>	day_mean <dbl>
94013	2	1
94101	1	3
94120	1	1
94130	2	1
95105	1	1

5 rows

We can find that the number of restaurants in each zipcode region is very low (only 1 or 2 restaurants) for these five zipcodes. Thus, their different levels of inspection frequency from the overall mean are resulted from the few samples in the zipcode region. Besides that, their low proportions in the whole data will not play a big role as confounders. Thus, locations or regions should not account for a major source of confounders.

2.4 Overall frequency of inspections over the time



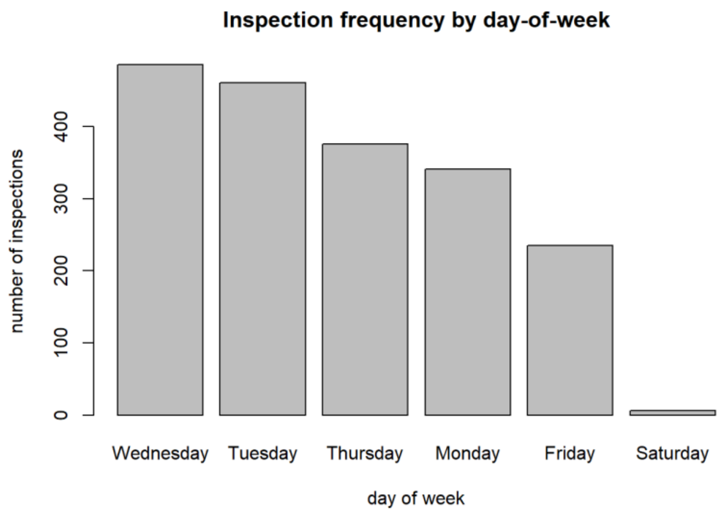
After investigating the confounding effect of the locations, we continue to inspect the seasonal effect. From the above plot, we can see that the inspection frequency does show different levels over the time. Instead of using the period of a certain month as a sharp cutoff for seasons, we want to find the more accurate boundaries of different seasons. To find these periods of different levels, we fit a piecewise constant function to the data, which uses the regression tree model. It partitioned the dates from 2016 to 2019 into seven seasons. And we can observe obvious differences of inspection frequency levels between each season. Thus, we should control for seasonal effect in the later permutation test. Because if in some specific season the inspection frequency is higher, then the probability of two nearby restaurants being inspected on the same day will increase. The frame of cutoffs for each season and the general levels of inspection frequencies are shown below.

start_date <date>	value <chr>	season <int>
2016-10-04	3.266	1
2017-03-09	1.938	2
2017-08-26	3.853	3
2017-12-12	3.211	4
2018-06-28	2.168	5
2018-12-30	3.619	6
2019-06-15	2.683	7

7 rows

2.5 Inspection frequency by day-of-week

After examining the confounding issues of seasonal trends, we also investigate the day-of-week effect. The plot of inspection frequency by day-of-week is shown below.



We can see that the inspection level of the dates are different in different day-of-weeks. This should be considered as another confounder in the permutation test. Based on their inspection frequencies, we group the day-of-week effect into three groups: Wednesday and Tuesday as group1, Thursday and Monday as group2, and Friday and Saturday as group3. Note that the level of Saturday is significantly lower than Friday. However, the number of cases on Saturdays is very low, thus to make things simple, we group Friday and Saturday into one group.

3. Pairwise distances and number of same_day inspections

3.1 Calculate pairwise distances

As the project asks us to demonstrate that nearby restaurants are likely to be inspected on the same day for efficiency. And we want to perform the permutation test to see whether there is evidence for this statement in the data. Therefore, we design the experiment to find the correlations of the pairwise distances of two restaurants and the number of same day inspections of these two restaurants.

First we need to calculate the pairwise distances. Because there are 997 unique restaurants in our data set, so there are 496506 pairs in total. As the location of a restaurant is shown in latitude and longitude, and we know that at San Francisco’s latitude, one degree of latitude corresponds to 69 miles, and one degree of longitude corresponds to 55 miles. So we can convert the coordinates from degrees into miles. Then we use the dist() function in R to calculate the pairwise distance matrix of all the restaurants, by which we get a distance matrix of dimension 997*997.

Then we create a pairwise distance data frame dist_df to represent each pair in a row and add a column to store their corresponding pairwise distances. After creating this data frame, we can further calculate the number of same day inspections for each pair.

3.2 Calculate number of same day inspections

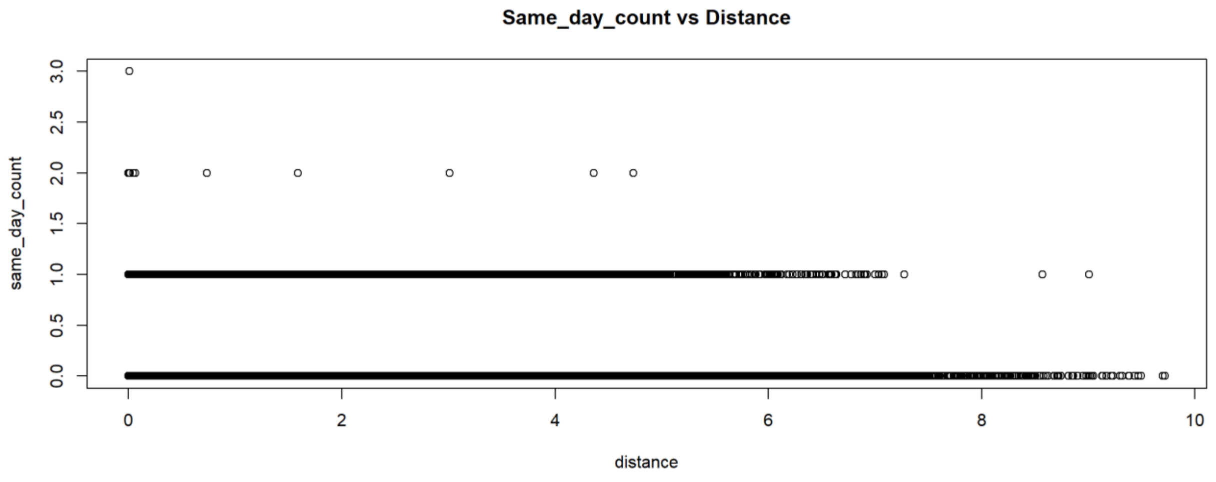
First we create a same_day matrix with dimension 997*997, where the entry x_{ij} means the number of days that restaurant i and restaurant j are inspected on the same day. So this matrix's diagonal entry $x_{ii}'s$ means the number of days inspected for restaurant i. And we find that there are 2956 days in total which different restaurants are inspected at the same day.

Then we can add the same_day_cnt columns in the data frame dist_df to show the number of same day inspections for each pair of restaurants. And by showing the table of the same_day_cnt, we can see that there are 2921 pairs are inspected at the same day one time, 16 pairs are inspected at the same day two times, and only 1 pair is inspected at the same day three times. So we know that the column same_day_cnt is a very sparse vector.

same_day_cnt	n
<dbl>	<int>
0	493568
1	2921
2	16
3	1

4 rows

We also plot the same_day_cnt against the pairwise distance. From the below plot, we can see that the closer the distance between two restaurants, the higher numbers of same day inspections. So we can first speculate that it is true that nearby restaurants are likely to be inspected on the same day for the sake of efficiency.



4. Permutation Test

4.1 Permutation test without controlling for confounders

(1) Only permute same_day_cnt, not the original dates

As mentioned above, we use the pairwise distance of two restaurants as vector X and the same day count of the pairs as vector Y to run a permutation test. The null hypothesis is that the same-day count of the pair is not correlated with the distance between the pair of restaurants. We permute the same_day_count in each permutation. The resulting p-value is 0.006644518, which is very significant. Thus, if we ignore the confounders, we will reject our null hypothesis. In other words, nearby restaurants are likely to be inspected on the same day.

However, in this method, if we want to compare with a method controlling for previously mentioned seasonal and day-of-week confounders , we will be unable to do that since the processing of aggregating the same-day inspection counts loses the daily level information. From our exploratory analysis, we find that different seasons and different day-of-weeks would result in different baseline in inspection frequency levels. If we fail to take these confounders into account, we will blend these different levels into our distance effect. In addition, because our X is the pairwise distance, it is not feasible to permutate within each time region to control for confounders related to time in this case. Thus, we change the subject of our permutation to original dates to get a more valid estimation of this problem.

(2) Permute the original dates

From the previous part, we know that it is not appropriate to only permute the vector of same_day_cnt, so in this part, we will permute the original dates of each restaurant and find new same_day_cnt for each pair in each permutation and then compute the corresponding p-value.

We set the number of permutations as 300, as it takes some time to run each permutation. Then for each permutation, we resample the data with date and set replacement is false. So we will get a new data set where the restaurants are inspected at different dates. Then we create the same day matrix and calculate the new same_day_cnt columns again using the same method discussed before. Finally we record the T_perms and calculate the p-values using the permutation method taught in class.

Using this method, while we ignore the confounders, the p-value is 0.01993355. We can see that this p-value is still significant, so if we ignore the confounders, we will reject the null hypothesis and say that nearby restaurants are more likely to be inspected on the same day. However, we also find that this p-value is larger than the p-value we calculated before, when permuting the same_day_cnt. I think this time our permutation is more reasonable and is valid to support our opinion.

4.2 Permutation test controlling for season, dayofweek_group

Now we control for both the day-of-week factor and the seasonal factor. We perform a task-specific permutation test, where we permute the original dates within the combinatorial groups of seasons and day-of-week groups. Again, for each group, we permute 300 times and calculate the p-values, which is 0.003322259. This p-value is much smaller than the previous one. Therefore, while considering the confounders, we will also reject the null hypothesis and conclude that nearby restaurants are more likely to be inspected on the same day, however, with a higher confidence level. Thus, on the other hand, without controlling for time-related confounders, we will underestimate this correlation.

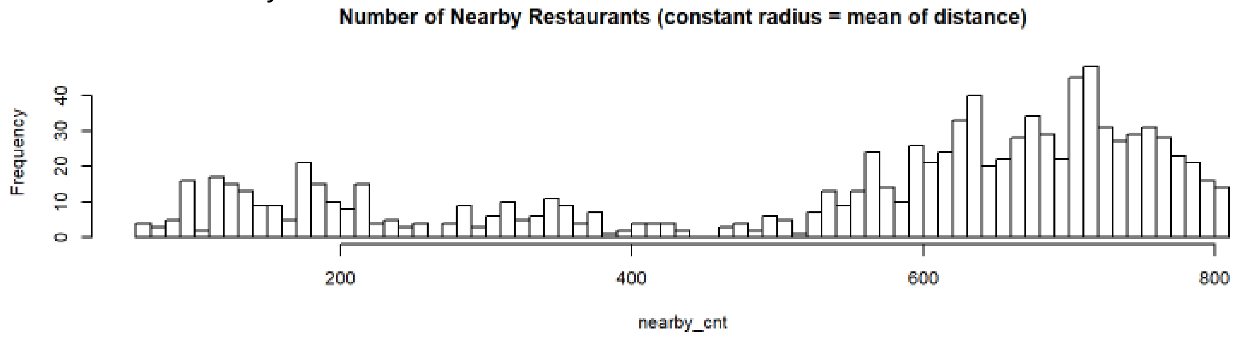
Problem 2

1. Find appropriate nearby restaurants

In problem 2, we are asked to find out whether it is true that if a restaurant was recently inspected and/or cited for a violation, it will influence the hygiene at nearby restaurants by improving their sanitation in fear of an inspection. We think it is reasonable, because when a restaurant is inspected and may get a low score for some risks, this may affect nearby restaurants, and be afraid that they are going to be inspected in the following days, so they may improve their sanitation after some inspections with other restaurants.

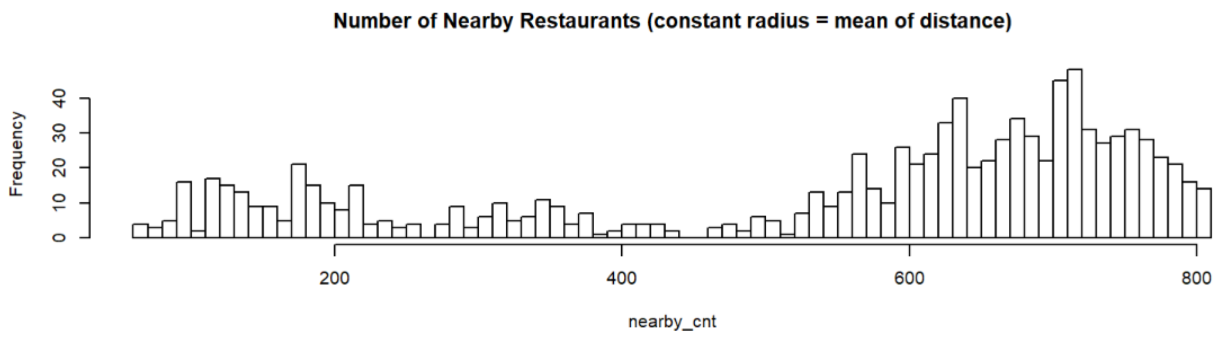
To prove that this statement is true, first we need to define how close they can be considered as nearby restaurants. Therefore, we define a nearby radius to find the nearby restaurants. First we summary the distance data frame, and we find that the mean distance of all these restaurants is 2.642 miles. From the latitude and longitude plot of these restaurants, we know that these restaurants are not uniformly distributed in San Francisco, most of them concentrate in the mid-northeast corner of the city. So if we want to find the nearby restaurants, we think it is inappropriate to choose a fixed distance as threshold to define whether the restaurants are nearby or not. We think the threshold radius should vary with the number of nearby restaurants.

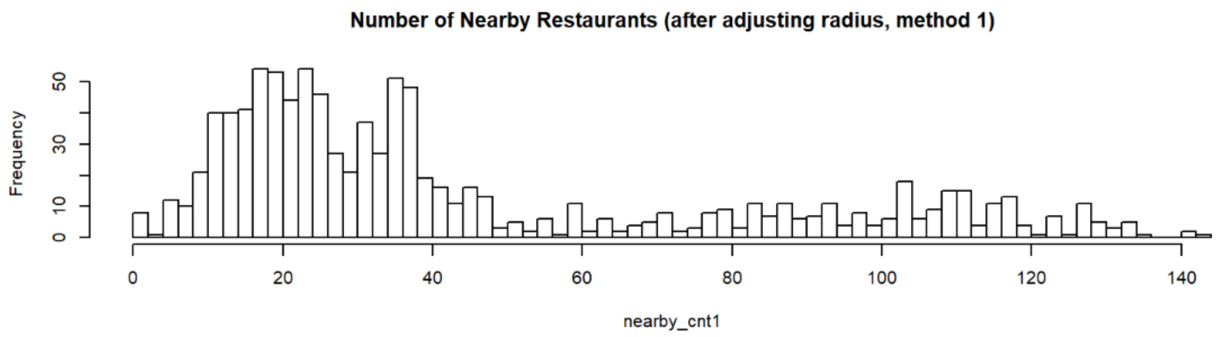
Therefore, first we set the threshold radius r equals the mean distance of all the restaurants, and find the restaurants whose distance is smaller than this radius, and we plot a histogram to see the number of nearby restaurants.



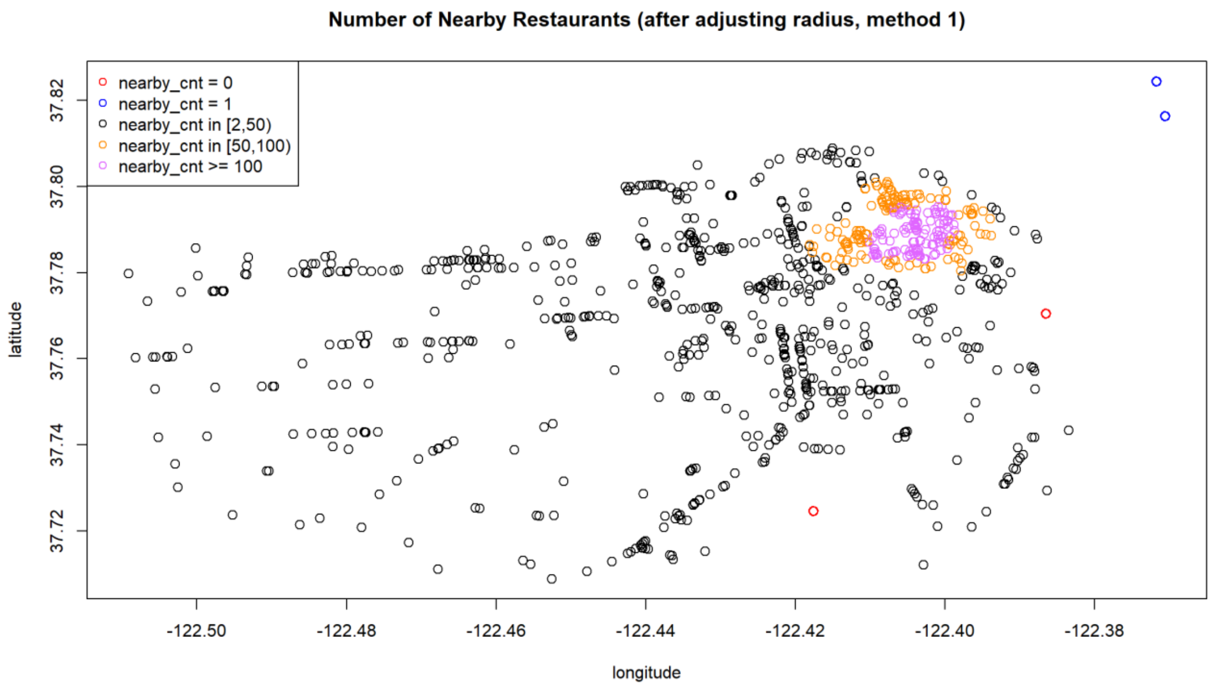
From this plot, we can see that if we choose to use the mean distance as threshold radius, there are so many restaurants with the number of nearby restaurants larger than 600, which we think is meaningless for our research. Therefore, we choose to adjust the radius and set different radius for different restaurants according to their first nearby count computed with r.

The first method is we first set the original radius r equals the mean distance of all restaurants. Then for each restaurant, let n be the number of restaurants in the circle centered by this restaurant and radius r, including the restaurant itself. After that we set a new radius new_r equals 4*r/sqrt(n). So the more number of nearby restaurants in the original radius,the smaller the new radius would be. And according to this adjustment, we can count a new number of nearby restaurants and plot them below in a histogram.





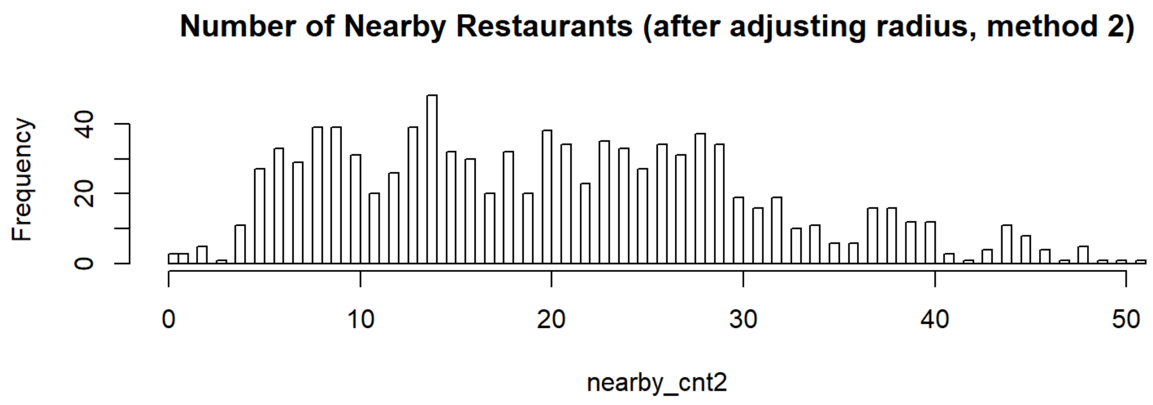
We can see that after adjusting the threshold radius corresponding to its intensity, most of the restaurants' nearby counts are smaller than 40, so it is more reasonable for us to discuss. We also plot the latitude and longitude plot of the restaurants and show their nearby counts below.



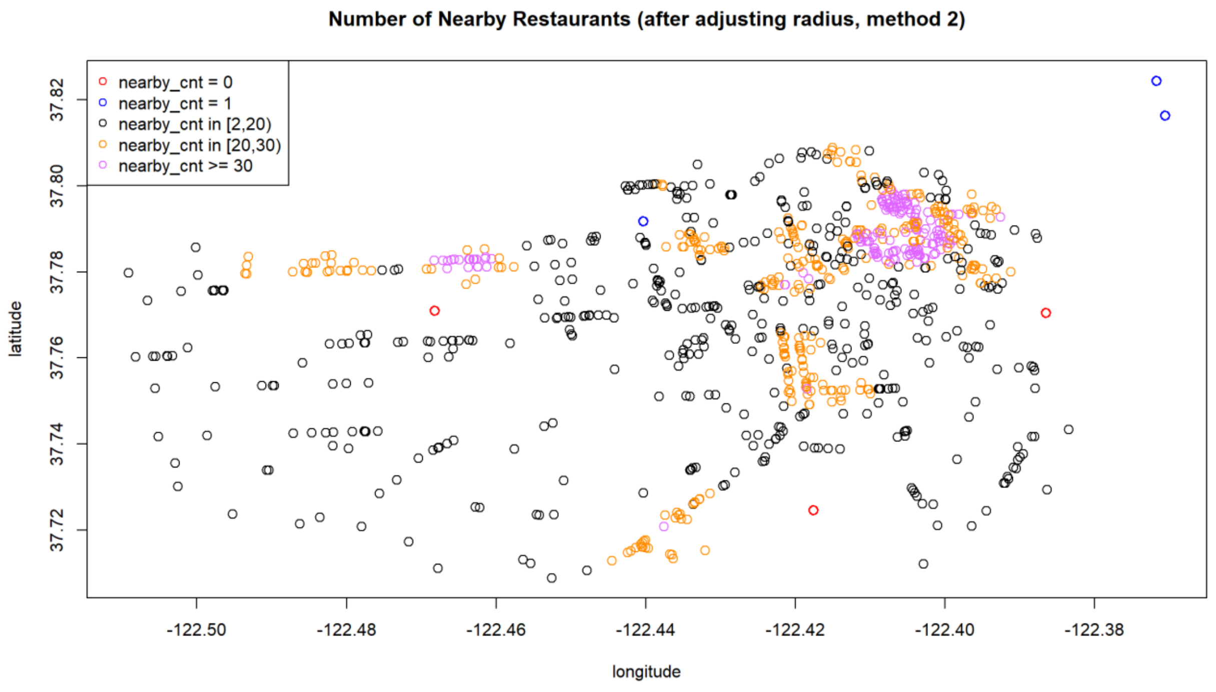
From this plot, we can see that there are only 2 restaurants that are not surrendered by other restaurants, and 2 restaurants have only 1 nearby restaurant (each other), and from the google map, we know that these two restaurants are opened in Treasure Island. Many restaurants have nearby counts ranging from 2 to 50. And when nearby counts are larger than 50, they are all concentrated in the mid-northeast part of San Francisco. So we can say that our method of defining the threshold radius is correct and appropriate for our research.

However, from the previous method 1, we can see that there are still a lot of points with nearby counts larger than 50. We still want to adjust the radius so that these points can have smaller number of nearby restaurants.

We define a new method of adjusting radius. First of all, similarly, we set the original radius r equals mean distance of all restaurants. Then we set new threshold radius new_r = 4*r/sqrt(n), just as what we did in method 1. However, we try to give more strength to shrink the circle. If the new count of nearby restaurants is smaller than 10, then the new_r stays. If the nearby counts is larger than 10 and smaller than 50, then we set new_r = 3*r/sqrt(n), if the nearby counts is larger than 50, then we set new_r = 2*r/sqrt(n). By using this new method of adjusting threshold radius, we can get a new and more appropriate number of nearby restaurants. And we plot it below.



We can see that by using the new adjusting method, almost all the nearby counts are under 50, and it is better for us to discuss whether our statement is true. We also plot the latitude and longitude plot of restaurants and mark them with different numbers of nearby restaurants.



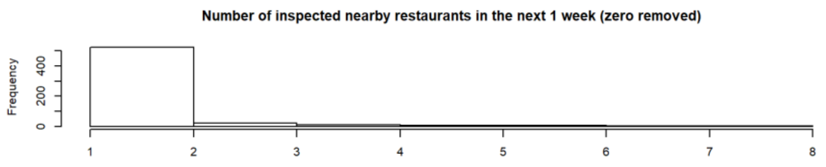
From this plot, we can also see that those special restaurants stay the same (none or only one nearby restaurant), and if the nearby counts is relatively high, we can see these restaurants are located in areas with high density. So we think this new method of adjusting threshold radius is more appropriate.

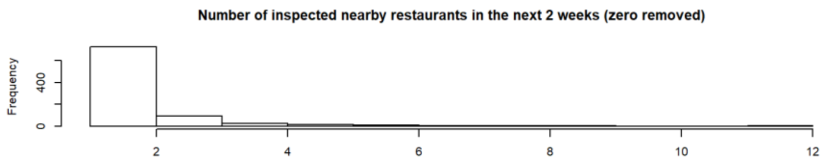
As we have decided how we can define the threshold radius, then we can get the nearby restaurants of each restaurant. We use method 2 to adjust radius, and define a matrix as restaurant_list, we will record every nearby restaurant which is smaller than the threshold radius for each restaurant. Then we can use this matrix for our further research.

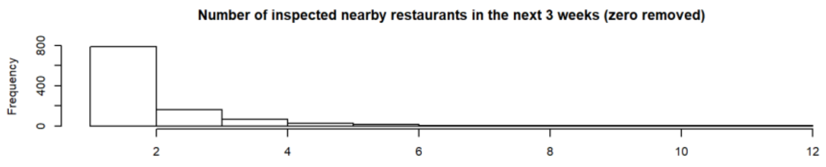
2. Define the future time period

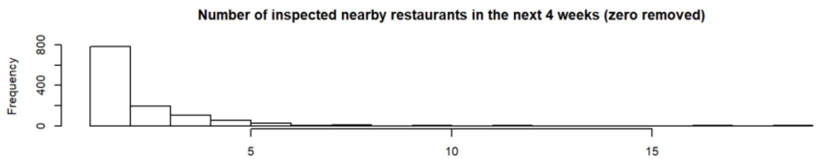
We would like to define a time period that captures the meaning of “recent”. If a restaurant was inspected one day, we define this date as the starting date and this restaurant as starting restaurant. Then we will extract all of the restaurants that were inspected after this starting date and before the starting date plus the time period. After that, we use our definition of nearby distance to get the nearby restaurants of the starting restaurant within our extraction and get their mean score. We also should get the mean score of the rest of the restaurants in our extraction. If we loop through all of the inspections, we will get two columns of mean-scores, one is for the nearby restaurants and one is for the other restaurants that were inspected within 28 days. Thus, the definition of time period is very important. If it is too short, we will not include enough restaurants in our extraction and this method will become unstable. However, if the period is too long, it will not be consistent with the definition of “recent”.

Thus, we investigate the number of inspected nearby restaurants in the next 1 week, 2 weeks, 3 weeks, and 4 weeks and the plots are shown below. From the plot, the count of inspected nearby restaurants is very low until we extend the time period to 4 weeks. For instance, if we set the time period equal to 1 week, most restaurants will have one nearby restaurants to be inspected. Thus, one month (4 weeks) should be the most suitable time period for our comparison.





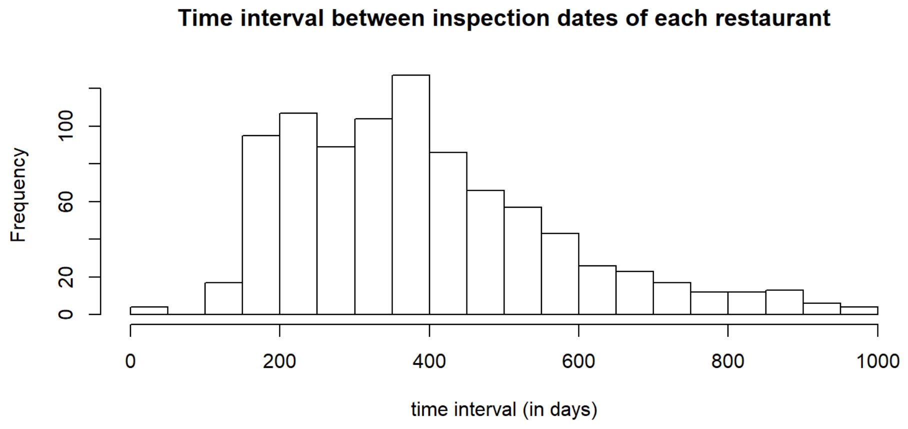




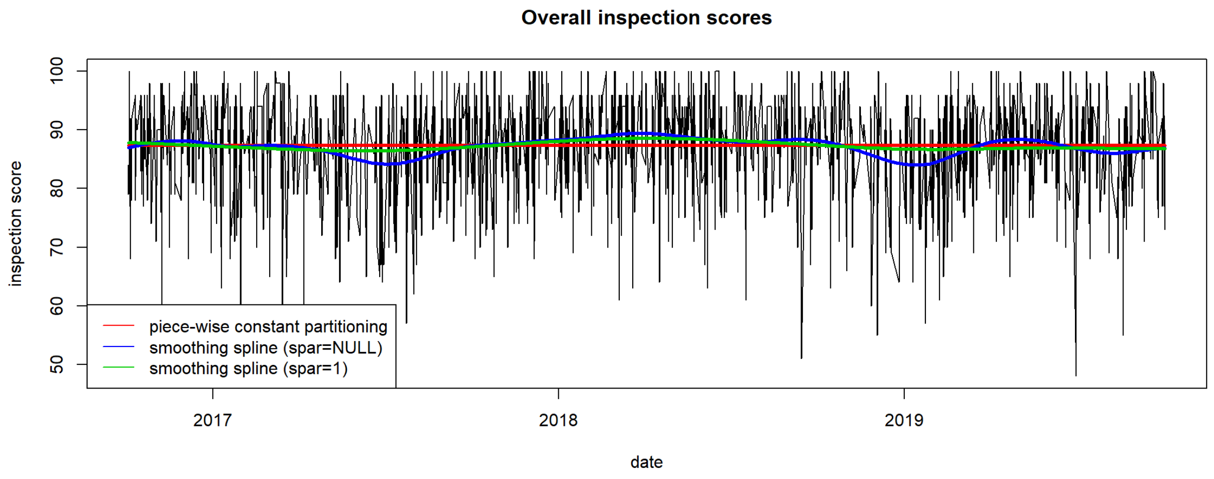
3. Whether the nearby restaurants improve their inspection score

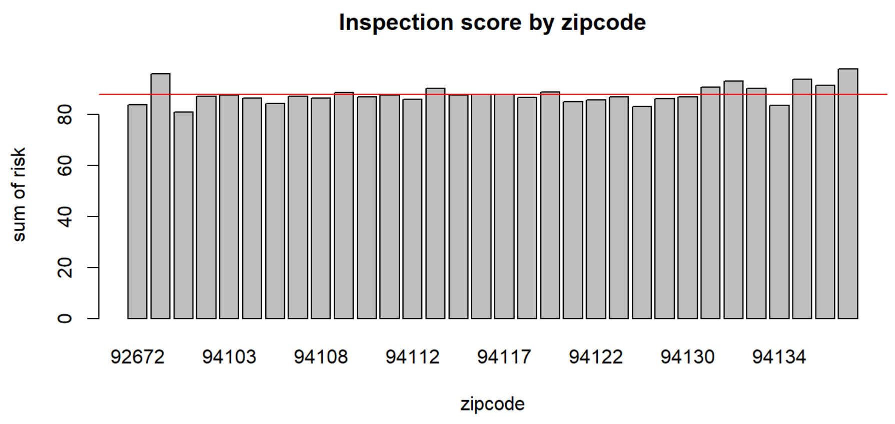
3.1 Exploratory Data Analysis

We first thought to investigate the difference of the nearby restaurants before and after an inspection. However, we calculated the time intervals for each restaurant between two inspections (see the below figure) and find that the general time interval is quite large. Therefore, it is not appropriate to compare the inspection score of a restaurant before and after the inspection of a nearby restaurant.



And then we investigate the trend of inspection score over the overall time period or the different zipcode regions. We can see in the below two plots that, the overall inspection scores has a slightly difference across the time (which can be also seen as approximately constant), but it does not show significant differences in different zipcode regions. Therefore, the overall inspection score is not changing significantly over all the time and all the regions. This provides us a way to compare the difference of the inspection score of the nearby restaurants. We can compare the inspection scores of the nearby restaurants in the next 4 weeks with those of the non-nearby restaurants in the next 4 weeks, since the mean of inspection scores of the non-nearby restaurants can represents a general inspection score level during that time.





3.2 Paired t-test and Results

After the missing values (no nearby restaurants were inspected in 30 days) were removed, there are 1191 pairs of values. We perform a two sided paired t-test to our data. Our null hypothesis is that the mean of the nearby restaurants that were inspected in 28 days was not equal to the mean of the other restaurants that were inspected in 28 days. The result is shown below.

```
        Paired t-test

data:  test_data$mean_near and test_data$mean_other
t = -2.4843, df = 1190, p-value = 0.01312
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8371719 -0.0983474
sample estimates:
mean of the differences
             -0.4677597
```

From the result, we find in fact the nearby restaurants' mean-score is significantly different with that of the other restaurants that were inspected in 28 days. However, nearby restaurants in fact get a lower score in average. To verify this conclusion, we perform a one-side test. The result is shown below.

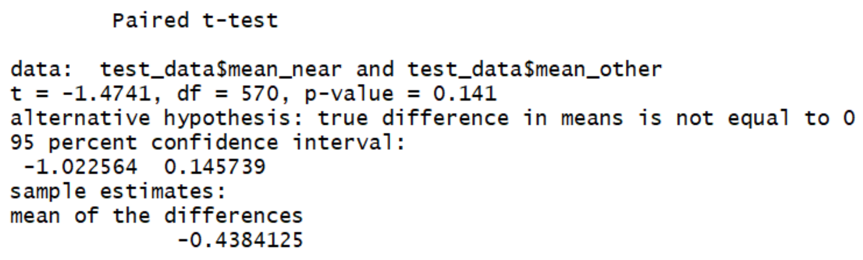
```
        Paired t-test

data:  test_data$mean_near and test_data$mean_other
t = -2.4843, df = 1190, p-value = 0.006559
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
       -Inf -0.1578131
sample estimates:
mean of the differences
             -0.4677597
```

Now our null hypothesis is that the mean of the nearby restaurants that were inspected in 28 days was less than the mean of the other restaurants that were inspected in 28 days. The P-value is 0.0066, which is very significant. Thus, based on our investigation, if a restaurant was recently inspected, the nearby restaurants will not improve their sanitation in fear of an inspection. In fact, the nearby restaurants even will get a lower score compared with other restaurants that were inspected within 28 days. However, the difference in score is not very big, which is 0.468 in average.

We also do the paired t test for different time periods, 1 week, 2 weeks, and 3 weeks. And we get their p-values and mean of the differences as below.

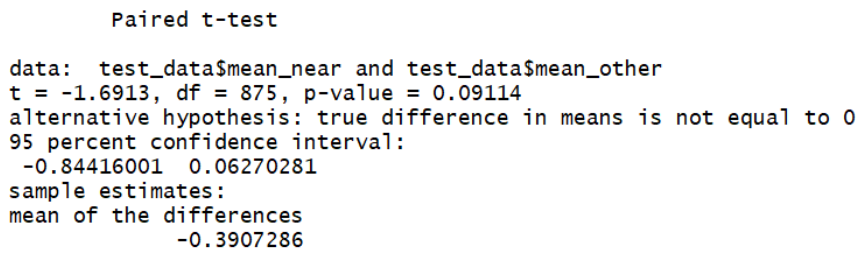
paired t test for future time period 1 week:



```
        Paired t-test

data:  test_data$mean_near and test_data$mean_other
t = -1.4741, df = 570, p-value = 0.141
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.022564  0.145739
sample estimates:
mean of the differences
             -0.4384125
```

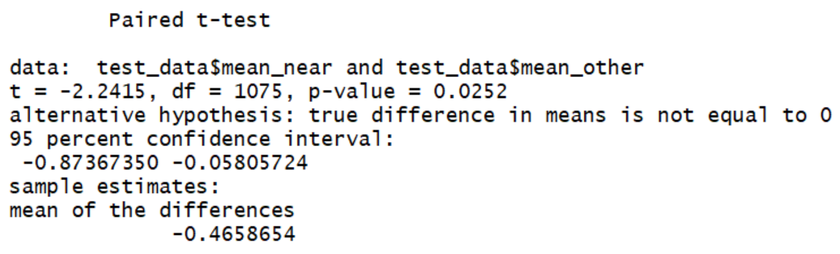
paired t test for future time period 2 weeks:



```
        Paired t-test

data:  test_data$mean_near and test_data$mean_other
t = -1.6913, df = 875, p-value = 0.09114
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.84416001  0.06270281
sample estimates:
mean of the differences
             -0.3907286
```

paired t test for future time period 3 weeks:



```
        Paired t-test

data:  test_data$mean_near and test_data$mean_other
t = -2.2415, df = 1075, p-value = 0.0252
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.87367350 -0.05805724
sample estimates:
mean of the differences
             -0.4658654
```

From these three tests, we can see that the p-value will be smaller if we let the future time periods to be longer. And they all get a minus mean of difference between the nearby restaurants and not nearby restaurants. This means that if a restaurant is inspected, then in a future time period, for example, 4 weeks, then its nearby restaurants will get a lower score than other restaurants which are not near this inspected restaurant. So the inspection will not make other restaurants to be vigilant. On the contrary, the inspection of one restaurant will make its nearby restaurants get a lower score! It seems not reasonable, but we can also explain this phenomenon. If a restaurant is inspected and has some risks, its nearby restaurants will get the information that this restaurant is inspected some days after the inspection date. Some of them may be vigilant and improve their performance. However, their improvement may not last long, and after some days, they may think that they are not going to be inspected any more, so they will be slack and back to normal, even perform worse. What’s more, some of the nearby restaurants may think that the inspectors are going to check only several restaurants in an area, so given that there is a nearby restaurant being inspected, there will be a smaller chance for them to be inspected. So instead of being vigilant, they may get a worse hygiene performance. This may also explain why the nearby restaurants get a lower mean scores than other restaurants in a future period if a restaurant is inspected.