Xtern FoodieX

1. Introduction

1.1 Background Research and Project Aims

In order to make FoodieX the best delivery service in town, the data science team is focusing its efforts on analyzing the data set to provide useful insights into the business.

1.2.1 Variables

There are total 10 variables. The data set contains Restaurant ID, Latitude, Longitude, Cuisines, Average Cost, Minimum Order, Rating, Votes, Reviews, and Cook Time.

1.2.2 10 Observations

The head eight observations are listed below:

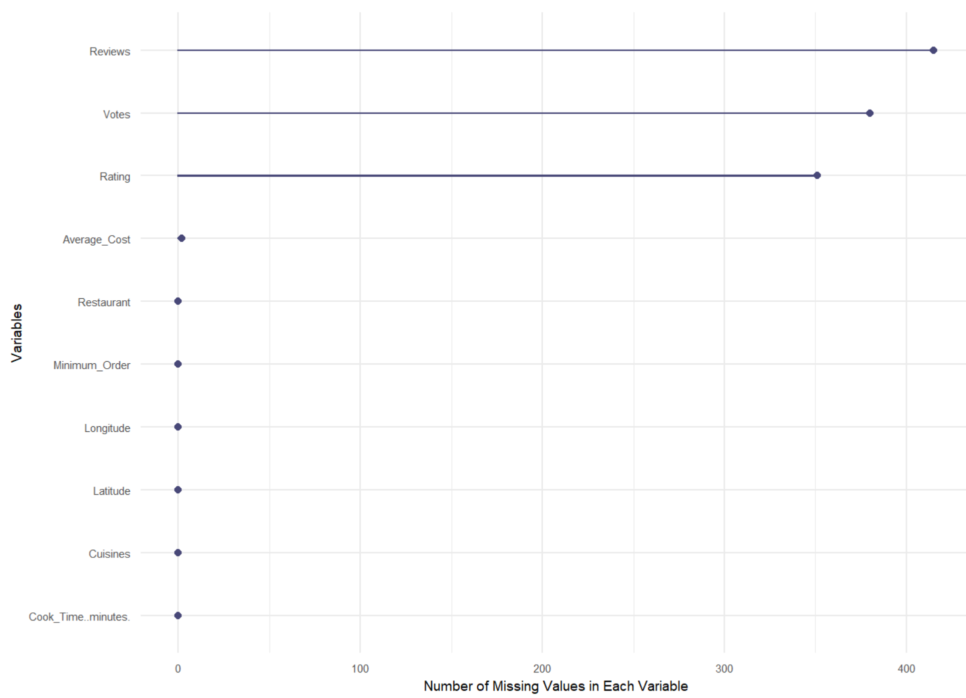
Restaurant	Latitude	Longitude	Cuisines	Average_Cost	Minimum_Order	Rating	Votes	Reviews	Cook_Time (minutes)
ID_6321	39.26261	-85.83737	Fast Food, Rolls, Burger, Salad, Wraps	20	50	3.5	12	4	30
ID_2882	39.77593	-85.74058	Ice Cream, Desserts	10	50	3.5	11	4	30
ID_1595	39.25344	-85.12378	Italian, Street Food, Fast Food	15	50	3.6	99	30	65
ID_5929	39.02984	-85.33205	Mughlai, North Indian, Chinese	25	99	3.7	176	95	30
ID_6123	39.88228	-85.51741	Cafe, Beverages	20	99	3.2	521	235	65
ID_5221	39.37044	-85.73952	South Indian, North Indian, Chinese	15	50	3.8	46	18	30
ID_3777	39.82181	-85.00558	Beverages, Fast Food	15	50	3.7	108	31	30
ID_745	39.28032	-85.14436	Chinese, Thai, Asian	65	50	4.0	1731	1235	45
ID_2970	39.26882	-85.60217	Mithai, Street Food	10	50	3.9	110	26	30
ID_3474	39.87452	-85.43996	Fast Food, North Indian, Rolls, Chinese, Momos, Mughlai	20	50	3.9	562	294	65

2. Summary Statistics and Data Visualization

2.1 Missing Values & Data Preprocessing

2.1.1 Missing Values

First We conduct basic data preprocessing. Missing values for dataset are shown in the histogram below.

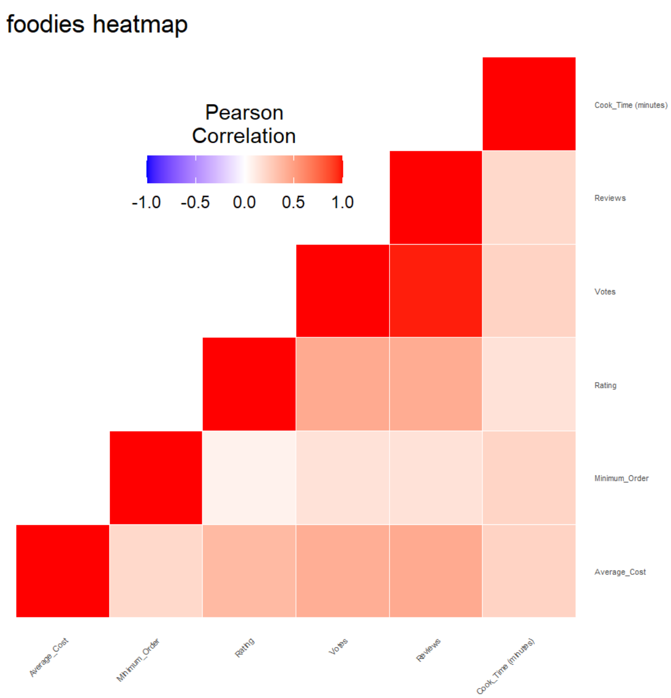


The plot above shows that the variables reviews, votes and rating has over 350 missing values and Reviews has the highest missing value. Due to the large number of missing values in dataset, completely delete missing values will result to a large amount of data loss. Thus, we use variable means to replace missing values.

```
## [1] "double"
```

2.1.2 Heatmap

Shown in below is a correlation map for the year 2010 data that describes the relationship between the different features. The heatmap below shows that all numeric variables have a positive correlation. Votes and reviews have especially high positive correlation.



3. Methodology

3.1 trying to identify the trending restaurants with your own scoring algorithm (can be as simple as the best rating or most votes or both!)

To understand the variables Ratings and votes better, I fist draw few histograms and plots. Since the variable Votes has so much bigger value than Ratings, I decided to have a new function called score which contain 1 rating and 0.01 votes since the plot shows that they have a positive relationship.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   10.00   20.00   20.03   20.00  150.00
```

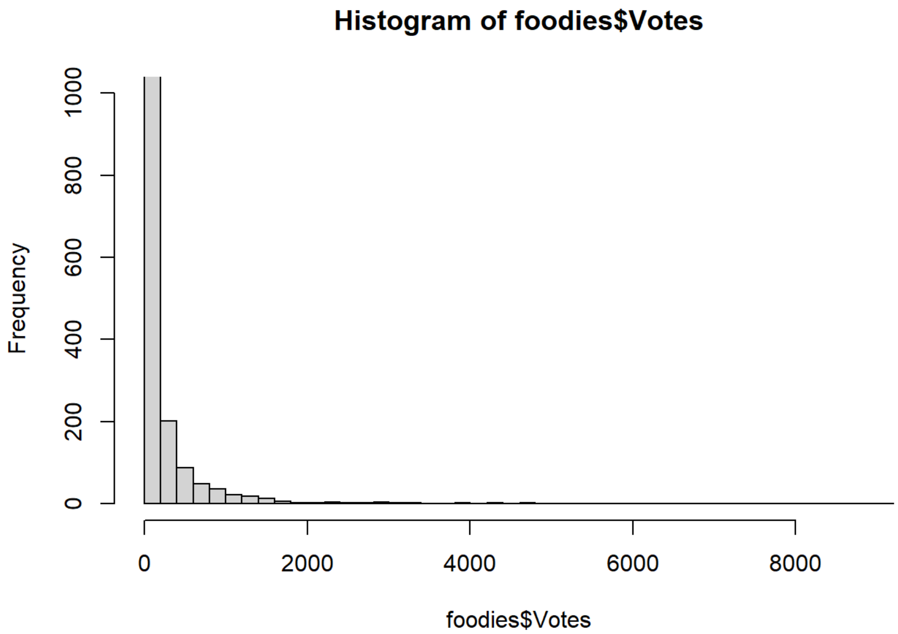
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   50.00   50.00   53.34   50.00  450.00
```

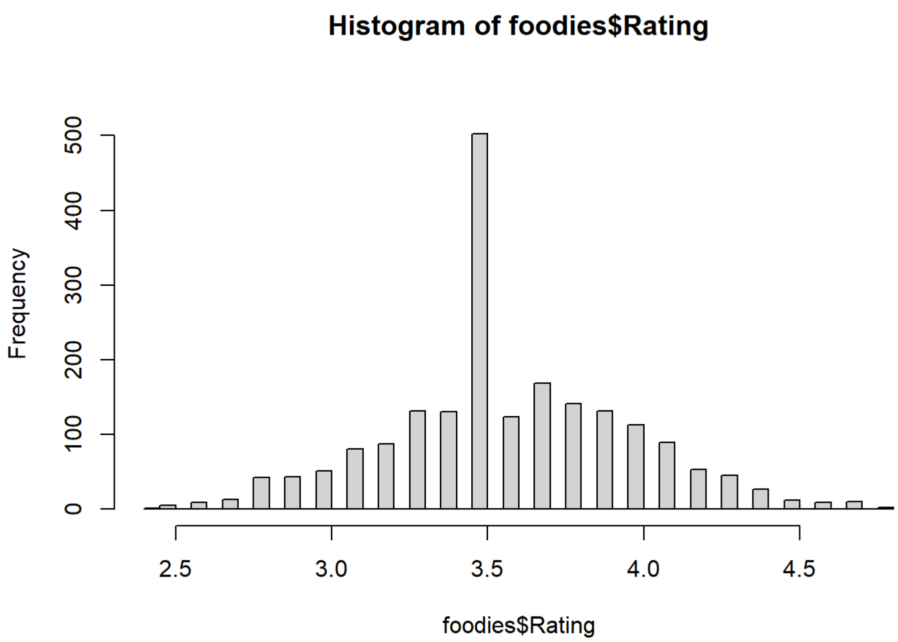
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.40    3.40    3.50    3.59    3.80    4.80
```

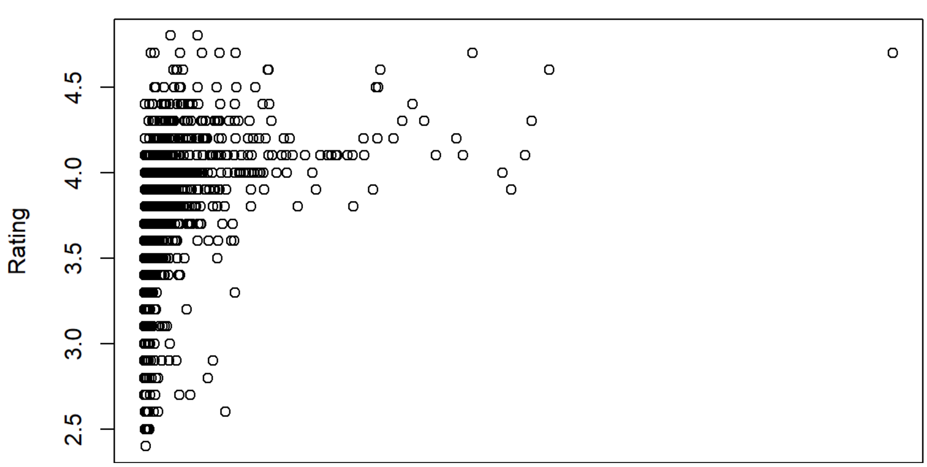
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0    12.0    36.0   209.1   175.0  9054.0
```

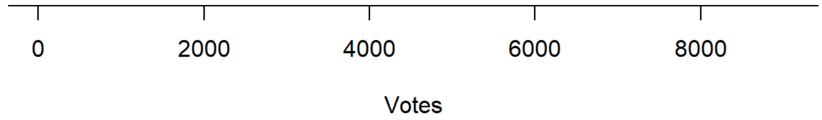
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       4      13     102      69    6504
```

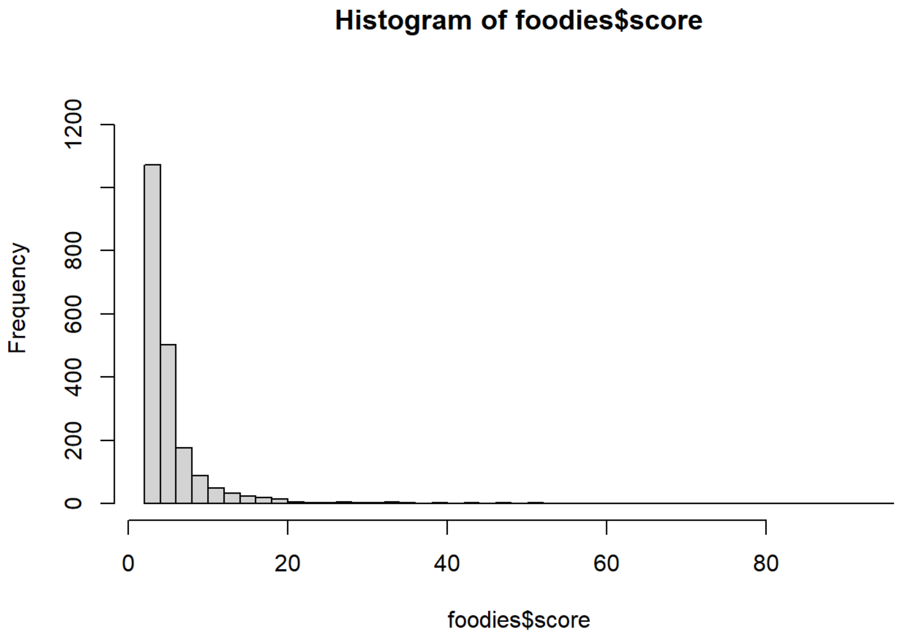
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   30.00   30.00   36.92   45.00  120.00
```


Histogram of foodies$Votes
Frequency
0
200
400
600
800
1000
0
2000
4000
6000
8000
foodies$Votes


Histogram of foodies$Rating
Frequency
0
100
200
300
400
500
2.5
3.0
3.5
4.0
4.5
foodies$Rating


Rating
2.5
3.0
3.5
4.0
4.5

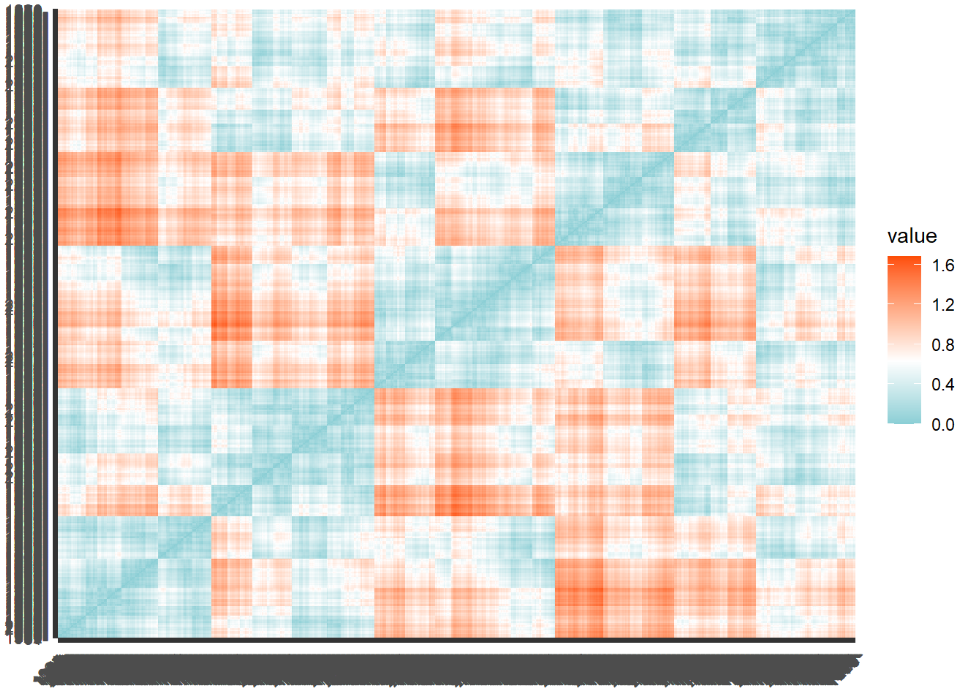





```
## # A tibble: 20 x 2
##    Restaurant score
##    <chr>      <dbl>
##  1 ID_1064     95.2
##  2 ID_1666     53.6
##  3 ID_2885     51.2
##  4 ID_2601     50.2
##  5 ID_6511     48.3
##  6 ID_4202     47.4
##  7 ID_4202     47.4
##  8 ID_2051     44.4
##  9 ID_13       42.7
## 10 ID_8087     42.0
## 11 ID_4606     39.4
## 12 ID_1947     38.2
## 13 ID_2041     36.9
## 14 ID_7753     35.5
## 15 ID_847      34.4
## 16 ID_6915     33.2
## 17 ID_2421     32.8
## 18 ID_7158     32.6
## 19 ID_4878     32.5
## 20 ID_988      31.6
```

I sorted the restaurant with scores and showed top 20 restaurant in the chart. The most popular restaurant is ID_1064, way above others

3.2 clustering restaurant locations to figure out the optimized FoodieX pick up zones



In k-means clustering, each cluster is represented by its center (i.e, centroid) which corresponds to the mean of points assigned to the cluster.

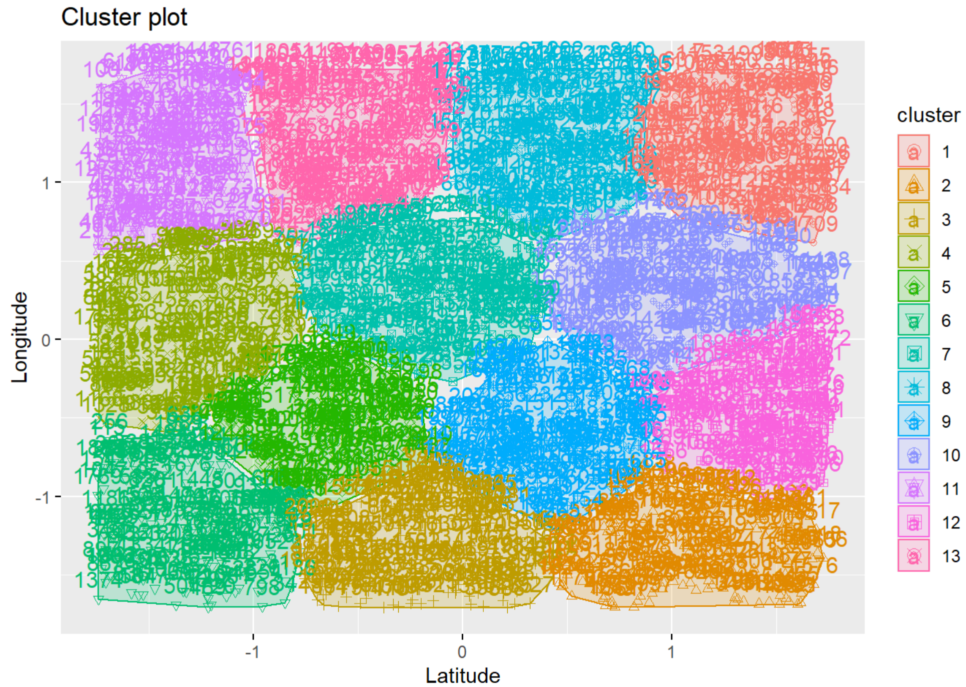
```
## [1] 79748.75
```

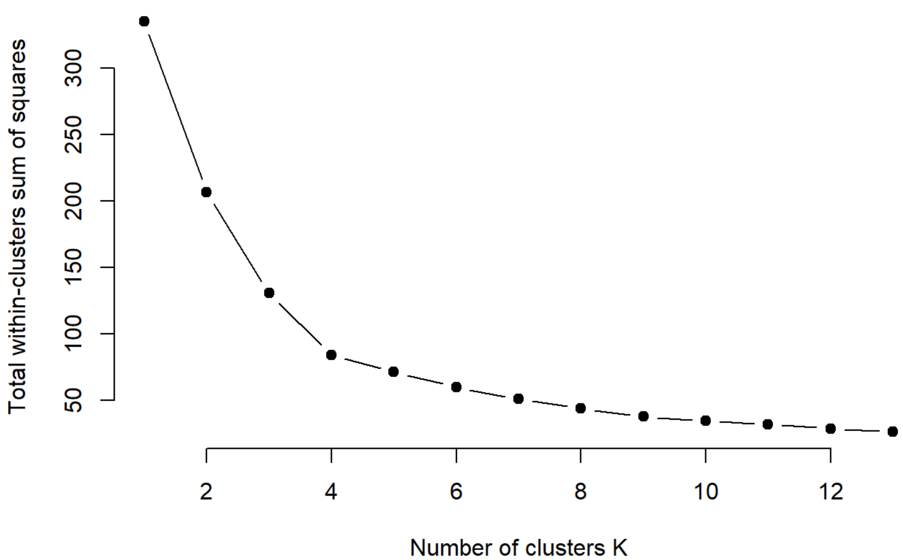
```
## [1] -172631.6
```

```
## List of 9
##  $ cluster     : int [1:2019] 6 12 13 11 10 5 1 13 5 10 ...
##  $ centers     : num [1:13, 1:2] 39.9 39.8 39.4 39.1 39.3 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:13] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:2] "Latitude" "Longitude"
##  $ totss       : num 335
##  $ withinss    : num [1:13] 1.91 1.98 2.06 2.39 1.5 ...
##  $ tot.withinss: num 26
##  $ betweenss   : num 309
##  $ size        : int [1:13] 151 145 159 161 148 150 174 169 154 158 ...
##  $ iter        : int 5
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
## K-means clustering with 13 clusters of sizes 151, 145, 159, 161, 148, 150, 174, 169, 154, 158, 133, 139, 178
##
## Cluster means:
##    Latitude Longitude
## 1  39.87993 -85.15918
## 2  39.81161 -85.89903
## 3  39.44453 -85.89422
## 4  39.12313 -85.49527
## 5  39.30849 -85.67209
## 6  39.12533 -85.85538
## 7  39.44961 -85.42331
## 8  39.61724 -85.14978
## 9  39.61377 -85.69011
## 10 39.79234 -85.42619
## 11 39.09934 -85.17254
## 12 39.90167 -85.65615
## 13 39.34513 -85.14455
##
## Clustering vector:
##    [1]  6 12 13 11 10  5  1 13  5 10  1 13  3 10  9  1  6 13  3  4 11 13  7  4
##   [25] 11  5  5  6 12  4  2  4  6  6  6 13  3 13  8  7  6  9  6  5  8 13 13 12
##   [49] 12  6 10  7  9  1  7  6  2 13  4  1  5  9  1 10  7  9  9  2  7 12  1  4
##   [73]  1  1  5 11  8  3  6 11  8  7 11  3 10  1  5  8  9  2  5 11 10 10 11  9
##   [97]  2  7  2 11  2 10  1 13  1  2  4  9  8  3  9 13 13  2  6  1 13 13  4  9
##  [121] 13  2  9  8 11 10  9  2 13 10 13 13 11  8  4  2  1  8 11  4  7  1 11  5
##  [145] 13  9  6  3  3  8  4  8  9  6  6 13 13  7  2  9 13  9  4 12  3  4  3  7
##  [169]  3  7  6 12  9  4 10  5  4 13 11  1  9 13  2  3  2 13  5  8  6  1  3  4
##  [193]  9  2  8  1  5 11  8  2  2 12  7  1  9  6  9  8  3  4 12 10 13  4 13  4
##  [217] 11  2  4  6  8  9 13  8  7  6  4  9  8  5 10 10  1 12  7  4  7  2  4 10
##  [241]  8  7  3  1  5 10 10  3 12  7  4  1 10  9  3  6  4  3  9  2  3 13  5 13
##  [265] 10 10  1  5 13 13  9  8  7 12 12 12 13 11  4  6 10 12 10  7  4  3  3 13
##  [289] 13 12 11  9 13  8 13  1  4  2  5  5  8  8 10  7  4  2 12 11  2  7  5  1
##  [313] 13  8  5  7  6  5  5  8  5 10  7 12  9 11  9  7  9  9 13  1 12  4 10  4
##  [337]  5  6 13  9  4  2 13 11  8  6 13  7  2  5  9  7  9  8  7 10 12  1  8 13
##  [361]  2  2  2  8  6  7  1 13  8 11 13  3  8  5  8  4  8  5  9  8  1  6  8  1
##  [385]  4  9 13  2 12 11  4  7 11  7  4  8  4 12 12  7 12  9  2 13 10 11 10  5
##  [409]  1  6 12 12  5 12 11  3  9  4  8  1  5 13 12  3  3  1  1  2 11 13  5  7
##  [433]  9  5  3  7  6 11  7  7  2  1 12  4 11  9  2  5 13  9  2  4  5 11  1  3
##  [457]  7  4  5  8 11  4  3  2 11  5  8 12  6  2  6 12  7  4 11  8  5  3  8  4
##  [481]  9  2 12  7 11 13  1 10 13  5  1  9  9  3  9  4  1 12 13  8  6  8  8  6
##  [505]  9  1  1  8  6  2  2  9  7  5  7 12 13  1  8  9  4 12  4  2  2  5  1  6
##  [529]  6  7 10  5  4 10  6  7  4  6  1 11 11  5 13  9  7  3 11 12  6 12  9  8
##  [553]  7  7  4 10  2  3  9  2  1  9  2  1 12  8  1  4  2  3  2 13  4  7 13  7
##  [577]  4  6  7  8  4  7 12 12  8  6 11  3 13  9  6  4  7  1  8  3  3 12  1 10
##  [601] 12 13 13  6 12  2  6  8 13  3  2  3  1  5  1 13  4  9 11  7  8 11  2  7
##  [625] 12  4  8  3 12  3  2 12  8 12  8  6  1  9  6  2 13  1  7  1  5 11  3  5
##  [649]  1  5  6  6  7  3  9  6 12  9 11  8  3  6 13  3  6  4  8  2  9 10  8  9
##  [673]  9  8  3 10 12  3  5  8  7  3 12 13  7 12  7  7  3  7  3  4  9 10  2  7
##  [697]  6  7 10  2  2  9  4  8  1  7  4  4  5 10  1  4 13  8  9  5  5  8  5  7
##  [721]  2  7  9 13  1 11  9  2  2  7 13  8 10  6  5 10 10  6  7  5  3  6  7  7
##  [745] 12 12  4 12  3  5  4 12  4  4  2  9  7  6  3  3 12  3 11  5 12 13 11 11
##  [769]  7  8 13  3  7 11  2 10  5  2  2  9  4  6  3  2  6 11 11  6 12  5 13  7
##  [793]  8 13  1  4  7  6  4 13  6 13 13  8 11  4 11  3  7  3 10  7  8  7 10 11
##  [817] 10  1  3  6  4  6 13  8 13  7  2  1 10  8  4  2  6  4 10  6  2  8  9  8
##  [841]  5  4  4 10  8  9 10 10 10 10 12 10 10  1  5  7  6  5 11  4  4  6  4 10
##  [865]  8  5  6 11  3  4  1  5 10  8 13 12  2 11  2  5  4  4 12  9  7  5 12  1
##  [889] 13  2 13  7  9  4  5  7  6  3  7  2 10  9  9  5  6  7  6 10  3  1  8  1
##  [913]  4 13 13  4 11  1  4  1  9 10 10  8  1  7  3  5 13  2  7 13  6  7  5 10
##  [937]  2  6  4 12  6  9  9  2  9  4 10  3  7  6  1  6  8  4  5  8  1  1  9  4
##  [961] 12  8  4  9 11  3  7 13  6  5 12  1 12 13 10  4 13 10  3  9  5  6  9 12
##  [985] 13  8 11  7  4  8 11  3  4  6  3 12  2  5  2  5 11 13  6 12  7  3  8  5
## [1009]  6  4  1 11  9  3  8  1  7  1  1 13 13  2 11 13 12  5  5  9 13  7  8  5
## [1033] 11 12  6  8 10 13  4  8 11  2 10 12 13  1  6  2  3  5  2  7  1 10 13  7
## [1057]  3 10  1  6  6  2  2 10  1  7  1  7  2  2  3 10  3 13  4  6 10 11  6  4
## [1081]  5 10  3 10  3  7 10  9  8  9  5  6  9  7 12  9  3  4  7  6  3 13  6 12
## [1105]  4 10  7  8  7  8  9  6 10  7  3 10 10 11 13 13  9  8 13  7  6  3  8  7
## [1129]  9  8  8 13  4  7  1  6  7 10  7 13  8  4 13 11  8  8 10  9  4 12  6  7
## [1153]  1  9  7  3  3  3  8  9 11 10  3  6  7  6  1  4  8  2  8  8  7  9  5  5
## [1177]  4 10  3  3  9  5  5  4 13  6 11  9 13  5  9 11  5 13 13 11 13 11  3 10
## [1201]  5  8 13  3  9  5  1  9 10  3 10  6  3 11  3  1  1  3 11  7 13  3  7  6
## [1225]  8  7  3  6 12 12 12  8  6  7  7 12  3 10  6 13  1  5  2  5  9  5  7  6
## [1249] 10 11  5 10  5  1  4  8  7  7  4 10 13 13  4  2  9  2 10  8  2  6  6  4
## [1273]  9  6  7  5 11 12  5 11 12  1  1  8  3  5 11  2  8 12  2  5  4  6  1 13
```

```
## [1297]  5  8  3 10  2  5  9  5  4  1 13  1  4  2  7 12 10 11  7  5  1 12 10  6
## [1321]  5  9 11 12 11  6  4  4  1 11 13  3 13  6 10  4  1 10  5  3  8  9  8  6
## [1345]  9 12  4  8  5  3 13  1  3  3  7 12  3  8  7  6  2 12 13 11 13  2  4  8
## [1369]  4  9  3  5  9  6  3 10  9  9 10  5  6  7 13  1  8  8  3 10 11  7  8  8
## [1393]  1 12  2  3  4  8  7 10 12  9 13 13  2 11 13  9 10  5  3 12  3 11  3  9
## [1417]  9  1  5  5 11 11  7  2  8 10  2  2 10  2  3 10  5 13 11  7  4 13  4 13
## [1441]  1  9  5  4  6  9 10 11  7 12  4  5 11 13 13  8  4 13  2 12  5 13  2  9
## [1465]  2 10 13  9  6  3  3  1 10  6  1  4  2 13  9 13  1 11  2 12  7 11  4  4
## [1489]  2  4  6 12 11  6  3 10  4  5  1  5  1  9  9 10  6  3  8  4 10  3  1  7
## [1513]  9  8 12  2  5  2  4  5 10  8 13 13  7  6 12 11  8  5 12  9 10  3  4  5
## [1537]  8 12 10  7  9 13  9  3  4  4  2  7  2 12  8  2  6  6  3  1  2  9  2 12
## [1561] 10  6  4 11  3 12 12  1  7  9  6 12 10  5 11  2 10 13  1 13 10  6 12  7
## [1585]  9  4 12  3  1 11 11  3  2  8  7 10 11  3  1 13  7 13 10  3  9 13  1 11
## [1609]  9 12  2 13  8  6  8 12  2 12  9  2  1 10 12  2  1  5  6 12 13  3  4  8
## [1633] 10  6  8 13  7 12 13 10  5 11  4 11  5 11  6 11  7  3 10  6 12 11  9  2
## [1657]  7  6  1  6 12  9 12 11  3  8 10  4  2  7  7 13  9 10  8  6  7  4  8 13
## [1681]  8  4  4  7  9  6  8  9  6  1 13 10 10 12  1  5 10  1  6 13 10  4  4  3
## [1705]  8  2 11 12  1 11 12  2 10  3 10 11  3  2  7  6  7  3 10  3  6 12  2  9
## [1729]  7 12  8  7  4  1  5  7  5  3  9  1  1 10  3  1  3  9 13  7  8  1  1  8
## [1753]  8  1 11  7  7  8  7  9 11  4  5 10 10 10 10 11  6  2  4 10 11 11  1 10
## [1777] 13  5  3  1  3  3  1 11  8 13  8  1  3  1  5  3 11  9  8  2  4  5  7  7
## [1801]  3 13 13  3 13 11  8  6  2 11 10  5  5 12  8  2  2  8  7 11  3 13  6 13
## [1825] 12  2  4  5  4  7 12 13  1 13 11 10  1 12 13 11  7  7  1  2  8  3 13 10
## [1849]  8 10  8  7 11  3  2  9  7 13  5 10  9  4  8  3  3 10  4 10  4 11  1  8
## [1873]  9  3  8  8  6 10 11  2  9 12  6  8 10  4  2  6  1 11 11  7 10 11 12  3
## [1897]  4 12  2  7  8  9 13  3  6  6  9  5 12  4  4  2  7 12 10  5 13 13  4  7
## [1921]  5  2 13  8  3  7  3  2  5 12  5  4  1  3  7 10  9 12  5  1 12  9 13 10
## [1945] 12  1 12  5 11  7 12  3  3  9  3  5  5  2  8 10  8 12 10 13 10  1  7  9
## [1969]  1 13 12  7  1  1 11 13  5  1  8  8 12  1  9 10  6 12 11  6  6  9 13  8
## [1993]  8 10  8  2  2  1 13  3  5 11  8  2  5  3 10  1 13  3 13 13 11  1  8  4
## [2017]  2  5  7
## 
## Within cluster sum of squares by cluster:
##  [1] 1.913501 1.983996 2.064139 2.387827 1.499562 1.946019 2.305848 2.269539
##  [9] 1.836914 2.245809 1.658518 1.462093 2.405514
##  (between_SS / total_SS =  92.2 %)
## 
## Available components:
## 
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```





3.3 estimating cook time based on restaurant info

first I need to initialize variables

And then I made two loops, the loop i using the restaurant ID and find that latitude and longitude. Second loop find the closest restaurant near by.

For example, I want to see the score for Restaurant 'ID_6321', this algorithm will show me that the score is 6.81 and cook time is about 45 mins

```
## [1] 6.81
```

```
## [1] 45
```

3.4 demonstrating your findings using a data visualization tool

please go to section 2 to see my data visualization(missing value chart and heatmap)

4 Reference

Foodies dataset is provided by TechPoint (https://drive.google.com/file/d/1DWleVQ00eG2rRTWNfEvTVWbUg5Aa_Usj/view)