

# HW2

Lily Monte

2025-02-11

github repo: [https://github.com/Lmonte695/plan372\\_homeworks](https://github.com/Lmonte695/plan372_homeworks) You should be a collaborator on this.

```
rm(list = ls())

setwd("C:/Users/Lily/Documents/collegefiles/PLAN 372/plan372_hmks/hw2")

library(tidyverse)

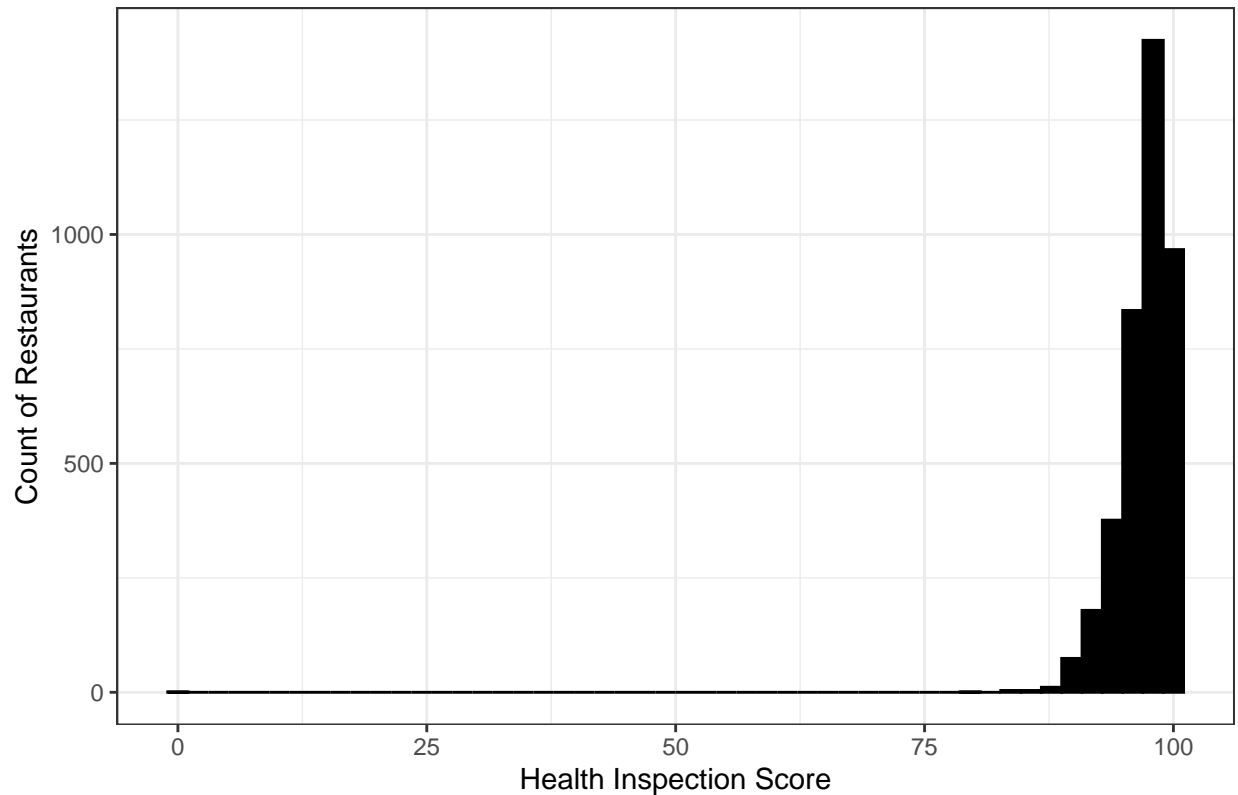
inspection_df = read_csv("restaurant_inspections.csv")
```

The above code sets up the document, changes the working directory, loads tidyverse, and assigns the csv file to the object “inspection\_df”.

## Question 1

```
ggplot(inspection_df, aes(x = SCORE)) +
  geom_histogram(color = "black", fill = "black", bins = 50) +
  labs(
    x = "Health Inspection Score",
    y = "Count of Restaurants",
    title = "Restaruant Inspection Scores in NC"
  ) +
  theme_bw()
```

## Restaruant Inspection Scores in NC



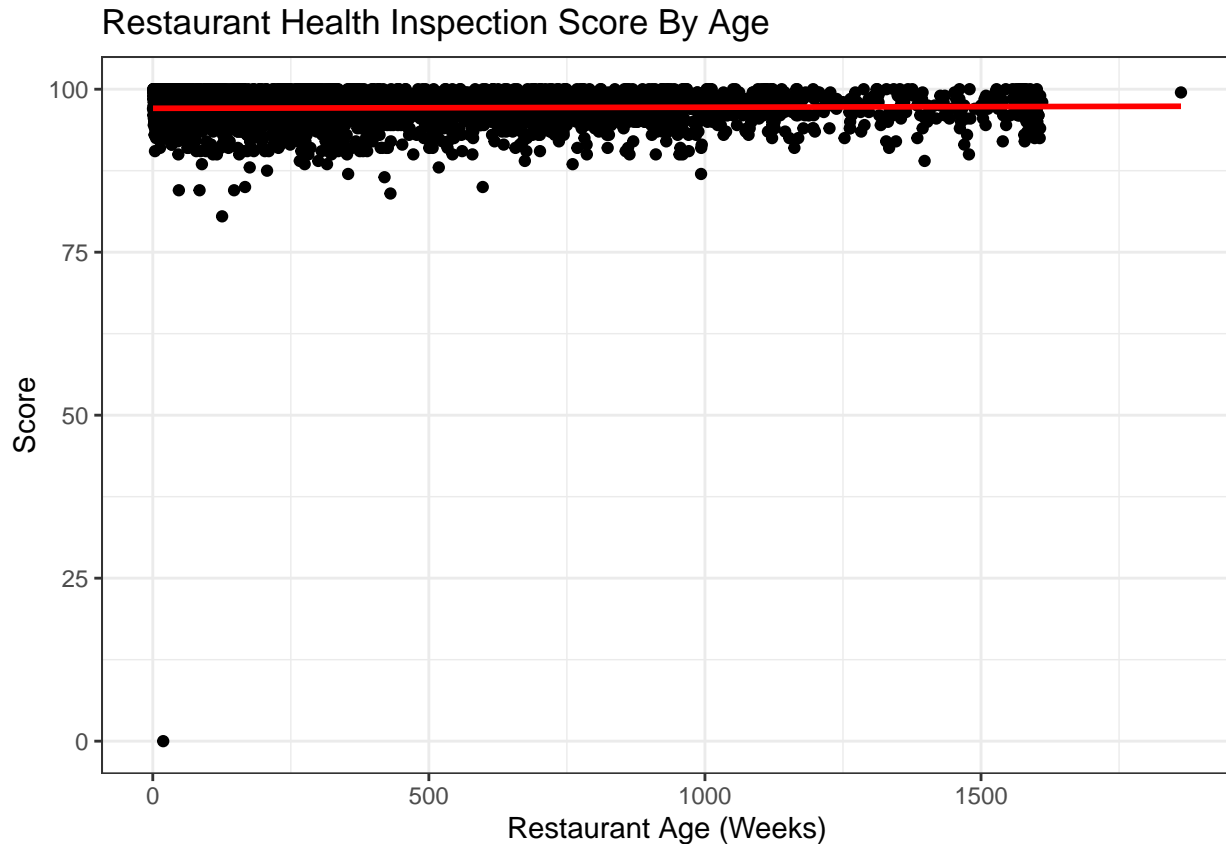
This code creates a histogram, showing that the spread of health inspection scores is mostly localized between 80-100 points. The code inside `geom_histogram()` sets the visual appearance of the bars, and `bins = 50` means that each bar represents all values that fall between 2 inspection scores.

## Question 2

To complete this question I conducted a linear regression to see if inspection scores varied by age of the restaurant, then plotted the relationship on a graph. The use of the `difftime` function here allows for the subtraction of the two columns and producing the result in “weeks”

```
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)   9.707e+01  7.140e-02 1359.556  <2e-16 ***
## restaurant_age 1.652e-04  1.091e-04   1.515    0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.807 on 3577 degrees of freedom
## (296 observations deleted due to missingness)
## Multiple R-squared:  0.0006415, Adjusted R-squared:  0.0003621
## F-statistic: 2.296 on 1 and 3577 DF, p-value: 0.1298
```

```
ggplot(inspection_df, aes(x = restaurant_age, y = SCORE)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(y = "Score", x = "Restaurant Age (Weeks)", title = "Restaurant Health Inspection Score By Age") +
  theme_bw()
```



The incredibly small slope of 0.00017 expected change in score between a brand new restaurant and a dining location one week older translates into about a 0.00884 point yearly increase. In other words, there is little to no relationship between age of a location that serves food and the score it receives. Of note is a single outlier which has a very low age and a score of 0, indicating the data probably needs further cleaning.

### Question 3

Because city is not a continuous variable, it will be difficult to perform a regression with it. One possibility is to perform a multivariate regression with dummy variables, but it seems more useful to plot a graph and examine the results visually, as well as numerically. First, though, I utilized base r commands to recode variables because mutate was causing issues that I could not solve.

```
unique(inspection_df$CITY)
```

```
## [1] "CARY" "RALEIGH" "KNIGHTDALE"
## [4] "CLAYTON" "FUQUAY VARINA" NA
## [7] "GARNER" "MORRISVILLE" "RESEARCH TRIANGLE PARK"
## [10] "RTP" "WENDELL" "Cary"
## [13] "APEX" "Apex" "WILLOW SPRING"
## [16] "HOLLY SPRINGS" "ROLESVILLE" "ZEBULON"
## [19] "Raleigh" "WAKE FOREST" "NEW HILL"
## [22] "FUQUAY-VARINA" "Zebulon" "Morrisville"
## [25] "Wake Forest" "Holly Springs" "ANGIER"
## [28] "Fuquay Varina" "NORTH CAROLINA" "MORRISVILLE"
## [31] "Fuquay-Varina" "HOLLY SPRING" "Garner"
```

```
inspection_df$CITY = toupper(inspection_df$CITY)
unique(inspection_df$CITY)
```

```
## [1] "CARY" "RALEIGH" "KNIGHTDALE"
## [4] "CLAYTON" "FUQUAY VARINA" NA
## [7] "GARNER" "MORRISVILLE" "RESEARCH TRIANGLE PARK"
## [10] "RTP" "WENDELL" "APEX"
## [13] "WILLOW SPRING" "HOLLY SPRINGS" "ROLESVILLE"
## [16] "ZEBULON" "WAKE FOREST" "NEW HILL"
## [19] "FUQUAY-VARINA" "ANGIER" "NORTH CAROLINA"
## [22] "MORRISVILLE" "HOLLY SPRING"
```

```
inspection_df$CITY[inspection_df$CITY == "RTP"] = "RESEARCH TRIANGLE PARK"
inspection_df$CITY[inspection_df$CITY == "FUQUAY-VARINA"] = "FUQUAY VARINA"
inspection_df$CITY[inspection_df$CITY == "HOLLY SPRING"] = "HOLLY SPRINGS"
inspection_df$CITY[inspection_df$CITY == "MORRISVILLE"] = "MORRISVILLE"
inspection_df$CITY[inspection_df$CITY == "NORTH CAROLINA"] = NA
```

```
unique(inspection_df$CITY)
```

```
## [1] "CARY" "RALEIGH" "KNIGHTDALE"
## [4] "CLAYTON" "FUQUAY VARINA" NA
## [7] "GARNER" "MORRISVILLE" "RESEARCH TRIANGLE PARK"
## [10] "WENDELL" "APEX" "WILLOW SPRING"
## [13] "HOLLY SPRINGS" "ROLESVILLE" "ZEBULON"
## [16] "WAKE FOREST" "NEW HILL" "ANGIER"
```

```
inspection_df = inspection_df %>%
  group_by(CITY) %>%
```

```
mutate(inspection_mean = mean(SCORE, na.rm = T)) %>%
ungroup

ggplot(inspection_df, aes(x = reorder(CITY, inspection_mean), y = inspection_mean)) +
  geom_point() +
  theme_bw() +
  theme(axis.title = element_text(size = 15),
        axis.text.x = element_text(angle = 90, vjust = .5)) +
  labs(x = "City",
       y = "Mean Inspection Score",
       title = "Mean Inspection Score by City")
```



The graph demonstrates that the lowest average scores were in Angier and Zebulon, while Research Triangle Park and New hill had the highest average scores.

#### Question 4

The following code first groups by the inspector variable and calculates the mean score for each of these groups, giving the mean score for each inspector. The `tibble()` function places the data into a small table-like dataframe, which is then viewed. The summary command allows for the viewing of key summary statistics about the data.

```
inspector_averages = inspection_df %>%
  group_by(INSPECTOR) %>%
  summarize(mean_by_inspector = mean(SCORE, na.rm = T)) %>%
```

```
ungroup %>%
tibble()

inspector_averages
```

```
## # A tibble: 39 x 2
##   INSPECTOR      mean_by_inspector
##   <chr>          <dbl>
## 1 Angela Myers      96.9
## 2 Angela Stocks     96.7
## 3 Brittney Thomas    98
## 4 Christy Klaus     96.3
## 5 Cristofer LeClair  97.7
## 6 Daryl Beasley     95.8
## 7 David Adcock      97.7
## 8 Dipatrimarki Farkas 97.8
## 9 Elizabeth Jackson  96.6
## 10 Ginger Johnson   97.6
## # i 29 more rows
```

```
summary(inspector_averages)
```

```
##   INSPECTOR      mean_by_inspector
## Length:39      Min.   :89.00
## Class :character 1st Qu.:96.18
## Mode  :character Median :97.02
##                  Mean    :96.78
##                  3rd Qu.:97.73
##                  Max.    :99.00
```

The inspectors all seem to have relatively high average scores, with a minimum of 89, a max of 99, and a median of 97.02. The harshest inspector was Thomas Jumalon, who averaged 89, and the most lenient was James Smith, who averaged 99

## Question 5

The following code creates 3 objects, then using the dplyr count function

```
facilitycount = inspection_df %>%
  count(FACILITYTYPE)
inspectorcount = inspection_df %>%
  count(INSPECTOR)
citycount = inspection_df %>%
  count(CITY)
citycount
```

```
## # A tibble: 18 x 2
##   CITY          n
##   <chr>        <int>
## 1 ANGIER        1
```

```
## 2 APEX 185
## 3 CARY 573
## 4 CLAYTON 4
## 5 FUQUAY VARINA 114
## 6 GARNER 133
## 7 HOLLY SPRINGS 107
## 8 KNIGHTDALE 81
## 9 MORRISVILLE 174
## 10 NEW HILL 2
## 11 RALEIGH 1895
## 12 RESEARCH TRIANGLE PARK 2
## 13 ROLESVILLE 24
## 14 WAKE FOREST 196
## 15 WENDELL 35
## 16 WILLOW SPRING 2
## 17 ZEBULON 50
## 18 <NA> 297
```

#### inspectorcount

```
## # A tibble: 39 x 2
##   INSPECTOR      n
##   <chr>      <int>
## 1 Angela Myers 138
## 2 Angela Stocks 52
## 3 Brittney Thomas 3
## 4 Christy Klaus 140
## 5 Cristofer LeClair 128
## 6 Daryl Beasley 16
## 7 David Adcock 71
## 8 Dipatrimarki Farkas 155
## 9 Elizabeth Jackson 137
## 10 Ginger Johnson 45
## # i 29 more rows
```

#### facilitycount

```
## # A tibble: 11 x 2
##   FACILITYTYPE      n
##   <chr>      <int>
## 1 Elderly Nutrition Sites (catered) 8
## 2 Food Stand 661
## 3 Institutional Food Service 46
## 4 Limited Food Service 1
## 5 Meat Market 93
## 6 Mobile Food Units 181
## 7 Private School Lunchrooms 13
## 8 Public School Lunchrooms 185
## 9 Pushcarts 39
## 10 Restaurant 2352
## 11 <NA> 296
```

It seems as if the sample sizes within all 3 of these groupings are highly varied. For example, only 1 limited food service and 8 elderly nutrition sites are recorded in the dataset, which are sample sizes far too small

to draw conclusions about. Furthermore, New Hill, which had particularly high mean scores, only had 2 restaurants in the sample - this indicates that the high scores may not be an accurate aggregate, the same is true for Angier, which had particularly low mean scores - but tested only one. For all of these cases, however, it would be important to understand how representative they are of the population; for example, does the distribution of food-service locations in the dataset mirror the distribution of food service in North Carolina at large?

## Question 6

The first bit of code in the following block adds a new column to `inspection_df`, using the `ifelse()` command to create a dummy that shows restaurants and changes all others to “other.” The next bit of code creates a new tibble called `facilityscores` that groups by restaurant & other, then takes the mean of both to find the average health inspector score for each category.

```
inspection_df = inspection_df %>%
  mutate(restaurant_dummy = ifelse(FACILITYTYPE == "Restaurant", "Restaurant", "Other"))

facilityscores = inspection_df %>%
  group_by(restaurant_dummy) %>%
  summarize(mean(SCORE, na.rm = T))

facilityscores
```

```
## # A tibble: 3 x 2
##   restaurant_dummy 'mean(SCORE, na.rm = T)'
##   <chr>              <dbl>
## 1 Other              98.1
## 2 Restaurant         96.7
## 3 <NA>              94.2
```

Restaurants seem to have a lower mean score than other food-service facilities, however only by about 1.5 points on average. This indicates slightly higher scores for non-restaurants, but the smaller sample size for non-restaurants may indicate that it is not representative of the population of non-restaurant food service locations.

## Question 7

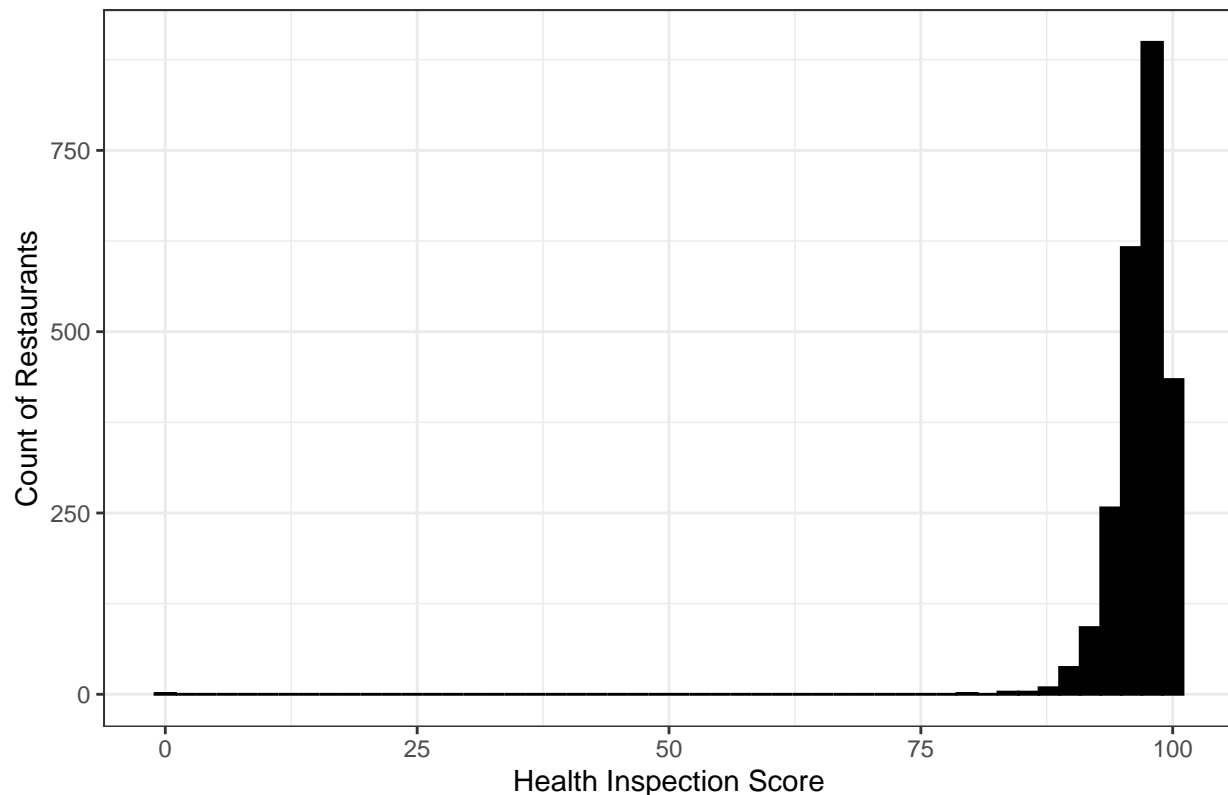
Essentially, the following codeblock copies the code of the previous question 1, but first filters by restaurant into a new dataframe, `restaurant_df`.

```
restaurant_df = inspection_df %>%
  filter(FACILITYTYPE == "Restaurant")

#Question 1 analysis with only restaurants
ggplot(restaurant_df, aes(x = SCORE)) +
  geom_histogram(color = "black", fill = "black", bins = 50) +
  labs(
    x = "Health Inspection Score",
    y = "Count of Restaurants",
    title = "Restaruant Inspection Scores in NC"
  ) +
  theme_bw()
```



## Restaruant Inspection Scores in NC



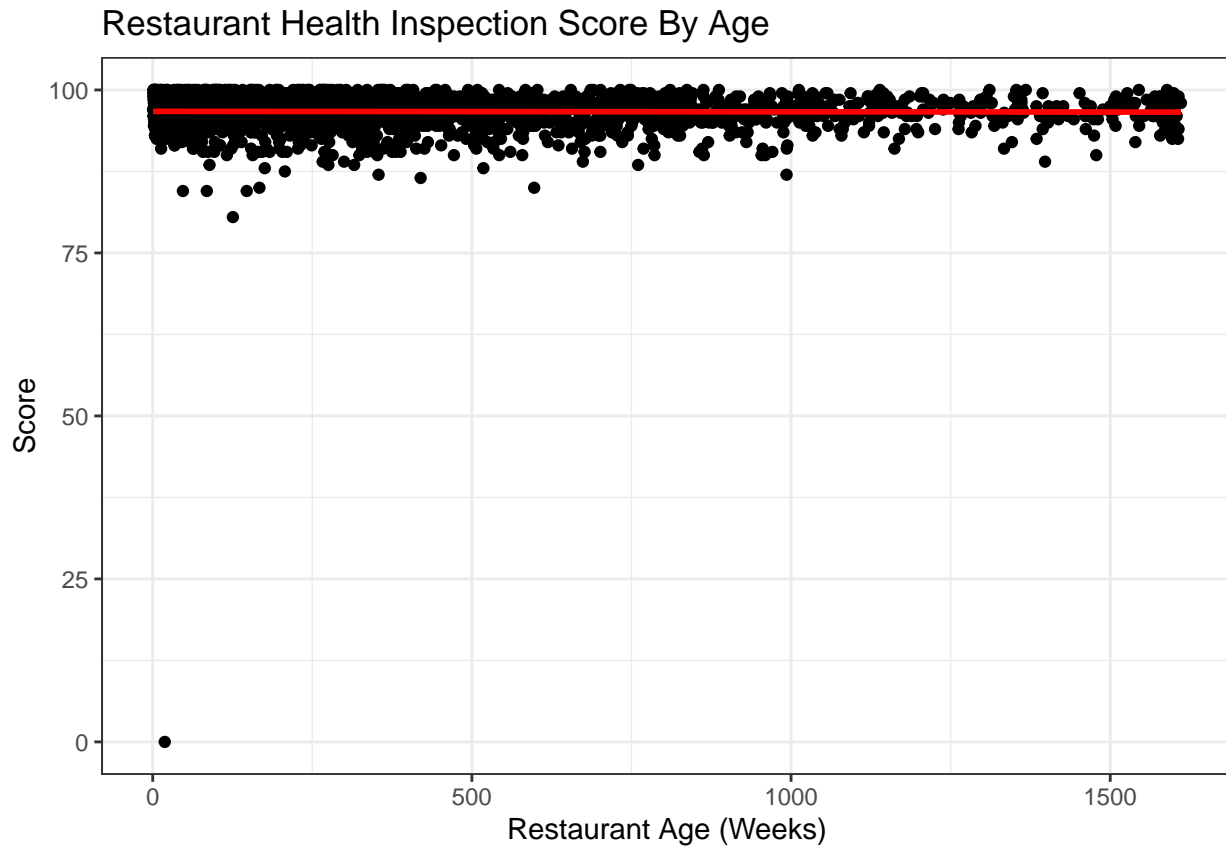
Part 2 of this question changes the outcome only slightly, removing an outlier of a particularly old location, but keeping the distribution mostly the same. The slope of the regression line changed slightly, but both round to 0 at 3 significant figures, so the change is negligible.

*#Question 2 analysis with only restaurants*

```
fitrestaurant = lm(SCORE ~ restaurant_age, restaurant_df)
summary(fitrestaurant)
```

```
##
## Call:
## lm(formula = SCORE ~ restaurant_age, data = restaurant_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -96.710  -1.184   0.325   1.800   3.394
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.671e+01  9.525e-02 1015.338  <2e-16 ***
## restaurant_age -6.629e-05  1.547e-04  -0.428   0.668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.08 on 2350 degrees of freedom
## Multiple R-squared:  7.813e-05, Adjusted R-squared:  -0.0003474
## F-statistic: 0.1836 on 1 and 2350 DF, p-value: 0.6683
```

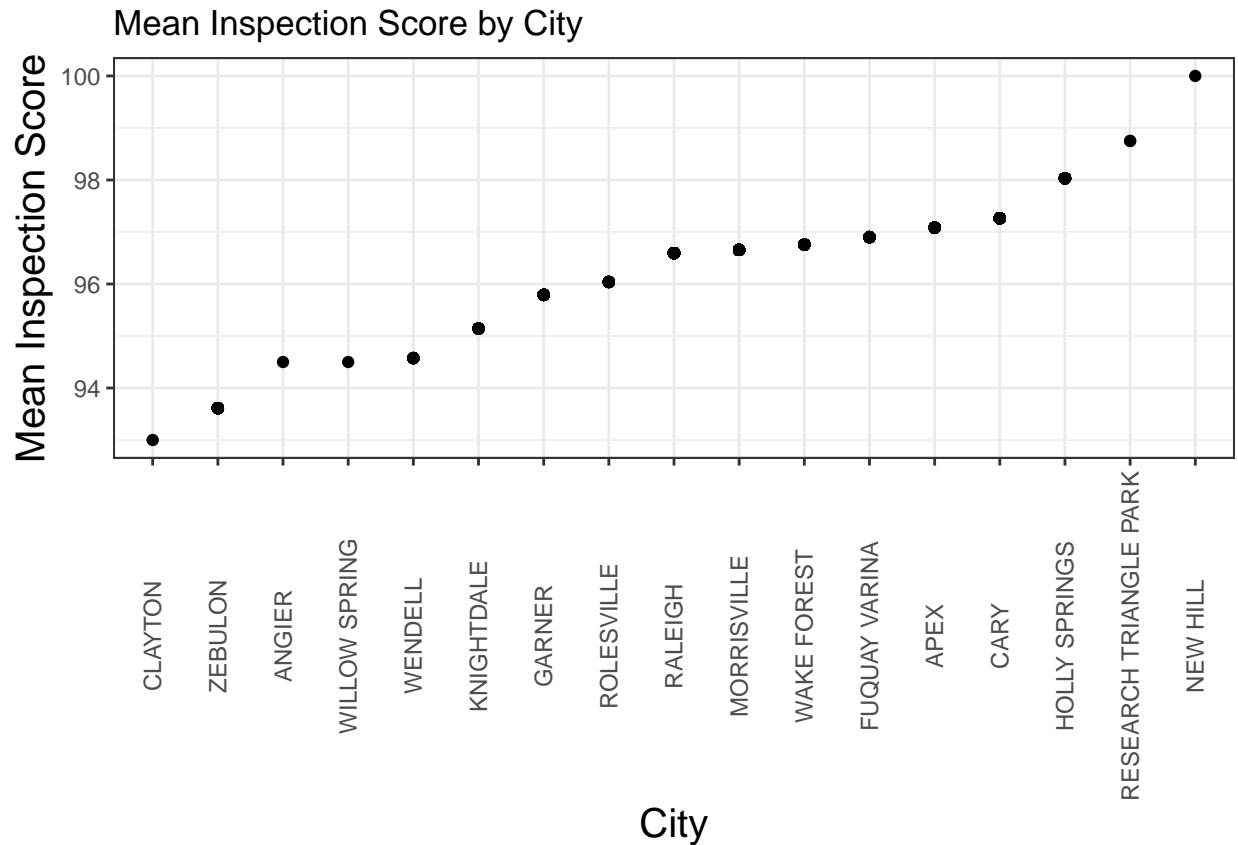
```
ggplot(restaurant_df, aes(x = restaurant_age, y = SCORE)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(y = "Score", x = "Restaurant Age (Weeks)", title = "Restaurant Health Inspection Score By Age") +
  theme_bw()
```



The code below does mostly the same as the code in question 3, albeit only using the `restaurant_df` dataframe. The code groups by city and calculates the mean for each city, then plots it on a graph.

```
#Question 3 analysis with only restaurants
restaurant_df = restaurant_df %>%
  group_by(CITY) %>%
  mutate(inspection_mean = mean(SCORE, na.rm = T)) %>%
  ungroup

ggplot(restaurant_df, aes(x = reorder(CITY, inspection_mean), y = inspection_mean)) +
  geom_point() +
  theme_bw() +
  theme(axis.title = element_text(size = 15),
        axis.text.x = element_text(angle = 90, vjust = .5)) +
  labs(x = "City",
       y = "Mean Inspection Score",
       title = "Mean Inspection Score by City")
```



The code below performs the same code as question 4 but utilizing `restaurant_df`. The mean's min and max do not change, indicating that utilizing only restaurants does not meaningfully impact average scores per inspector. Of note, however, is the presence of one less row than in question 4, indicating that at least one inspector never inspected a restaurant.

```
# Question 4 with restaurant only
inspector_averages = restaurant_df %>%
  group_by(INSPECTOR) %>%
  summarize(mean_by_inspector = mean(SCORE, na.rm = T)) %>%
  ungroup %>%
  tibble()
```

```
inspector_averages
```

```
## # A tibble: 38 x 2
##   INSPECTOR      mean_by_inspector
##   <chr>          <dbl>
## 1 Angela Myers    96.7
## 2 Angela Stocks   96.2
## 3 Brittny Thomas   98
## 4 Christy Klaus   95.9
## 5 Cristofer LeClair 97.1
## 6 Daryl Beasley   95.4
## 7 David Adcock    95.9
## 8 Dipatrimarki Farkas 97.7
## 9 Elizabeth Jackson 95.7
## 10 Ginger Johnson  97.6
```

```
## # i 28 more rows
```

```
summary(Inspector_averages)
```

```
##   INSPECTOR      mean_by_inspector
## Length:38      Min.   :88.00
## Class :character 1st Qu.:95.90
## Mode  :character Median :96.75
##                Mean   :96.37
##                3rd Qu.:97.59
##                Max.   :99.00
```

This final code block once again reproduces question 5's code using the `restaurant_df` dataframe. Naturally, only restaurants are examined so only these locations show up in the table. Overall counts are of course reduced, but seemingly maintain a similar distribution among the two other categories. Overall, the results for only restaurants do not seem particularly different from the total results.

```
facilitycountrest = restaurant_df %>%
  count(FACILITYTYPE)
inspectorcountrest = restaurant_df %>%
  count(INSPECTOR)
citycountrest = restaurant_df %>%
  count(CITY)
citycountrest
```

```
## # A tibble: 17 x 2
##   CITY              n
##   <chr>            <int>
## 1 ANGIER              1
## 2 APEX             108
## 3 CARY              406
## 4 CLAYTON             1
## 5 FUQUAY VARINA      76
## 6 GARNER             93
## 7 HOLLY SPRINGS      80
## 8 KNIGHTDALE         49
## 9 MORRISVILLE     144
## 10 NEW HILL             1
## 11 RALEIGH          1193
## 12 RESEARCH TRIANGLE PARK  2
## 13 ROLESVILLE        13
## 14 WAKE FOREST       133
## 15 WENDELL            20
## 16 WILLOW SPRING        1
## 17 ZEBULON            31
```

```
inspectorcountrest
```

```
## # A tibble: 38 x 2
##   INSPECTOR          n
##   <chr>          <int>
## 1 Angela Myers    104
```

```
## 2 Angela Stocks      36
## 3 Brittney Thomas    3
## 4 Christy Klaus      100
## 5 Cristofer LeClair  72
## 6 Daryl Beasley      12
## 7 David Adcock        8
## 8 Dipatrimarki Farkas 118
## 9 Elizabeth Jackson   80
## 10 Ginger Johnson     35
## # i 28 more rows
```

```
facilitycountrest
```

```
## # A tibble: 1 x 2
##   FACILITYTYPE      n
##   <chr>          <int>
## 1 Restaurant     2352
```