HW2_Julia_Bright

This file available at: https://github.com/JBrighttt/plan372_hmks/tree/main

Question 1

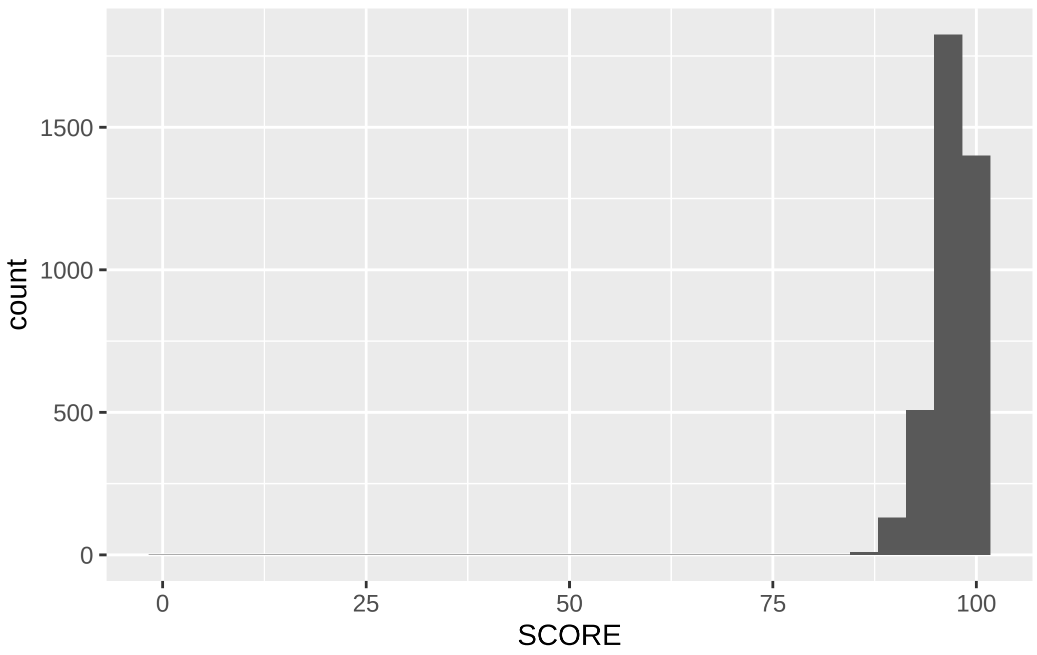
```
unique(ri$SCORE)
```

```
 [1]  97.0  96.0  98.5  90.5  97.5  98.0 100.0  99.5  95.0  88.0  90.0  96.5
[13]  91.0  91.5  94.0  93.5  99.0  94.5  92.5  95.5  93.0  92.0  89.5  88.5
[25]  84.5  89.0  85.0  87.0  85.5  80.5  87.5  86.5   0.0  84.0
```

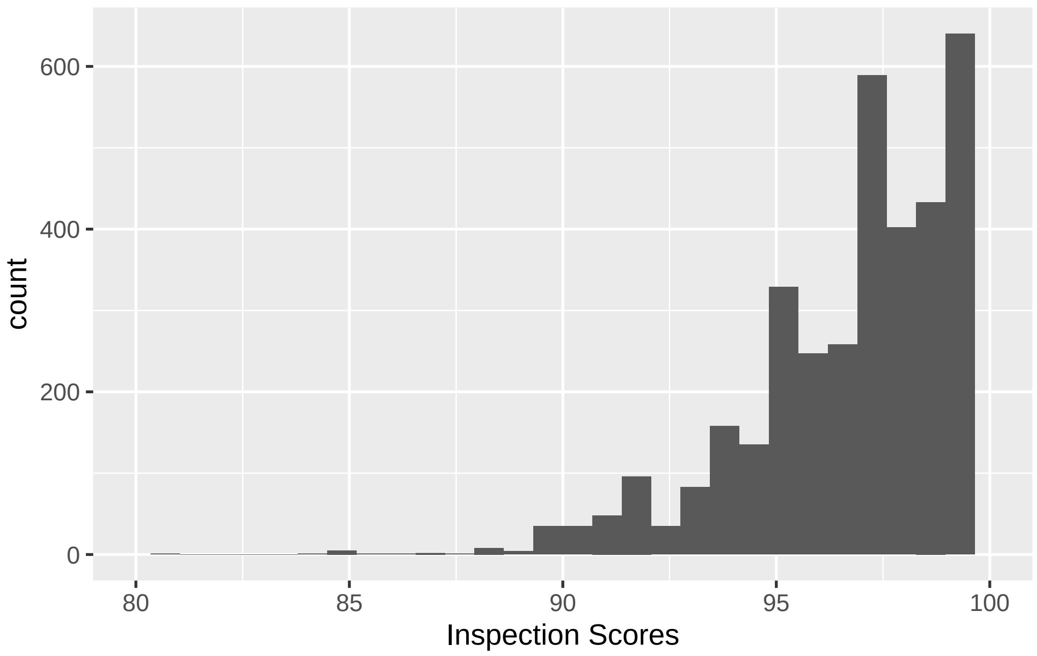
```
table(ri$SCORE) #i see there is only a single 0 score
```

```

   0 80.5   84 84.5   85 85.5 86.5   87 87.5   88 88.5   89 89.5   90 90.5   91
   1    1    1    3    2    1    1    2    1    3    5    4    1   34   35   48
91.5   92 92.5   93 93.5   94 94.5   95 95.5   96 96.5   97 97.5   98 98.5   99
  49   47   35   83   76   82  135  177  152  247  258  305  284  402  433  330
99.5  100
 310  327
```

```
ggplot(data=ri, aes(x=SCORE)) + geom_histogram()
```

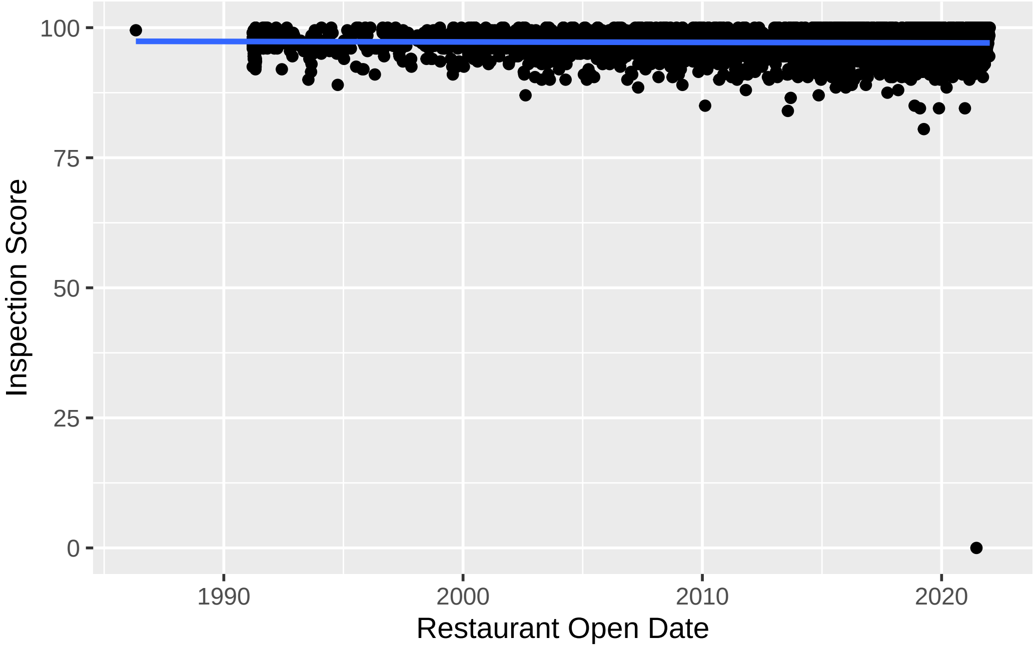


```
#would be useful to set xlim to see a more visually instructive distribution
ggplot(data=ri, aes(x=SCORE)) + geom_histogram() + xlim(80,100) +
 ↪  xlab("Inspection Scores")
```

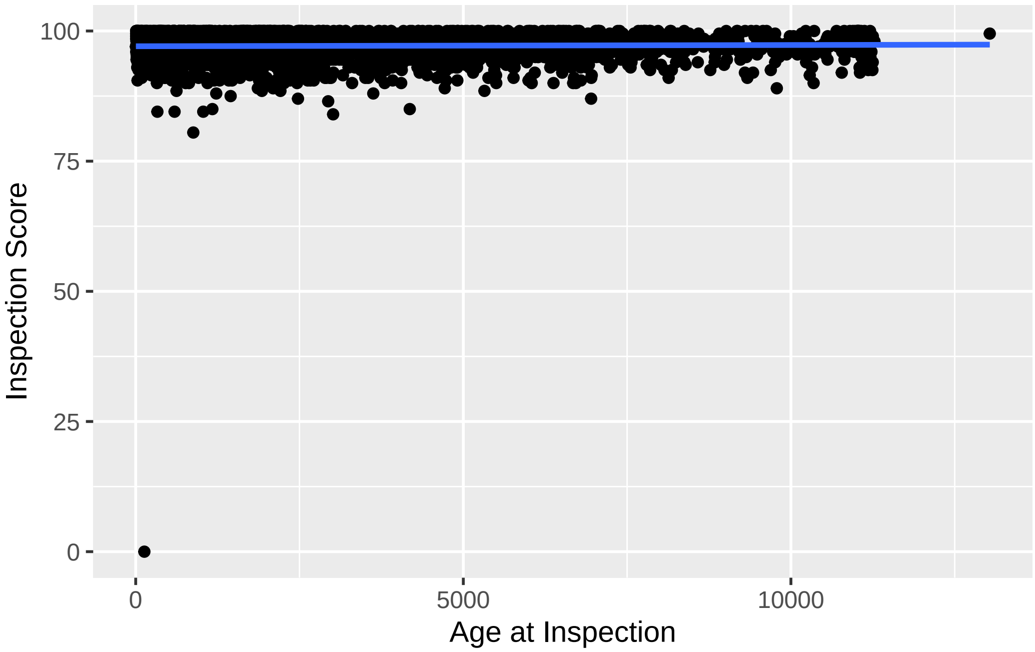


Question 2

```
ri$RESTAURANTOPENDATE <- as.Date(ri$RESTAURANTOPENDATE, format = "%Y/%m/%d")
 ↪  #formatted the open date column to function as date values

fit1 <- lm(SCORE~RESTAURANTOPENDATE, data = ri) #created linear regression of
 ↪  inspection scores as a function of open date
ggplot(data=fit1, aes(x=RESTAURANTOPENDATE, y=SCORE)) + geom_point() +
 ↪  geom_smooth(method="lm") + labs(x="Restaurant Open Date", y="Inspection
 ↪  Score") #plotted linear regression, no apparent relationship between
 ↪  restaurant open date and inspection scores
```



```
#let's check age instead of open date
ri$DATE_ <- as.Date(ri$DATE_, format = "%Y/%m/%d") #formatted inspection date
 ↪  column

ri$age_to_inspection <- ri$DATE_ - ri$RESTAURANTOPENDATE # create new
 ↪  variable determining the age of facility at date of inspection
ri$age_to_inspection <- as.numeric(ri$age_to_inspection) #format it as
 ↪  numeric to function properly in linear model
fit2 <- lm(SCORE~age_to_inspection, data=ri)
ggplot(data=fit2, aes(x=age_to_inspection, y=SCORE)) + geom_point() +
 ↪  geom_smooth(method="lm") + labs(x="Age at Inspection", y="Inspection
 ↪  Score") #basically the same but flipped due to taking the age rather than
 ↪  open date. No discernable relationship.
```



Question 3

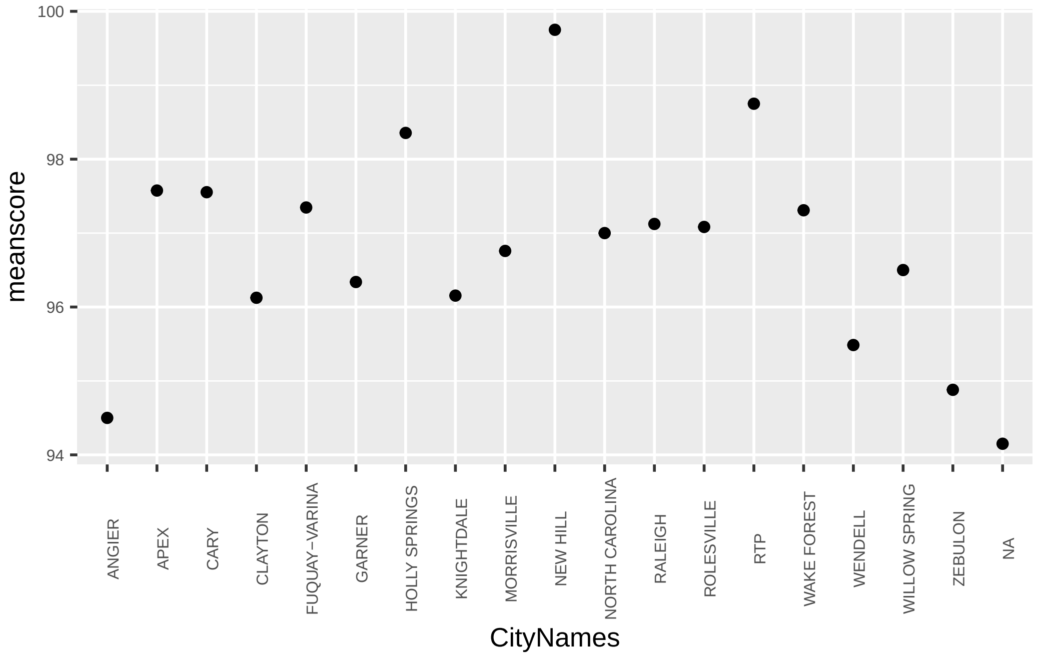
```r
unique(ri$CITY)
```

```
 [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"         "MORRISVILE"
[31] "Fuquay-Varina"          "HOLLY SPRING"           "Garner"
```

```
ri$CITY <- str_to_upper(ri$CITY) #converted all to upper case, removes 10 of
 ↪  the duplicates
unique(ri$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] "MORRISVILE"              "HOLLY SPRING"
```

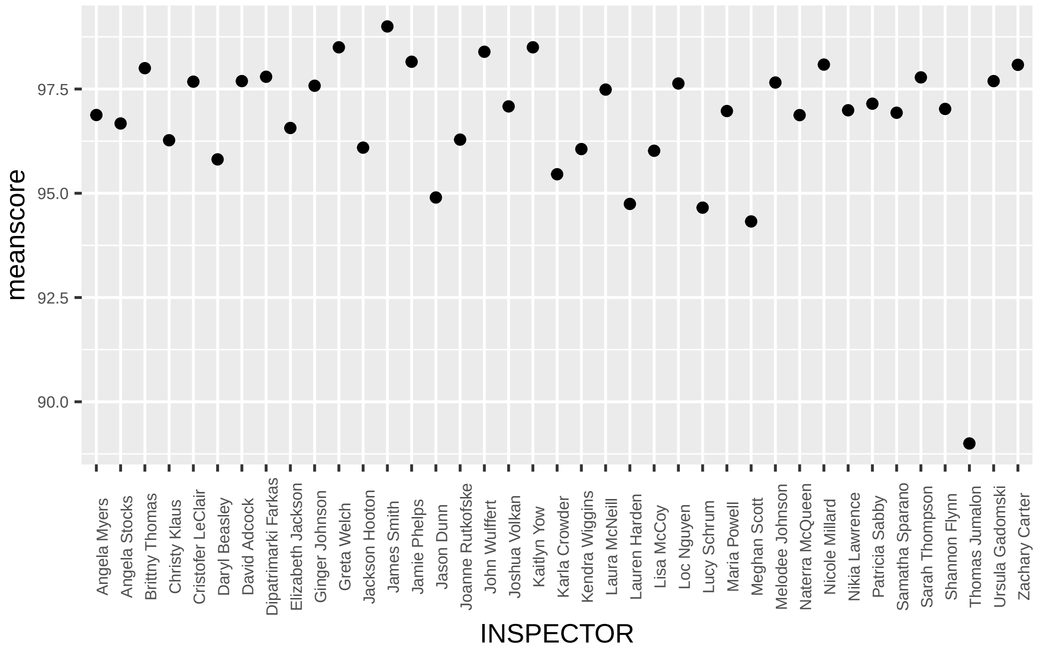
```
ri$CityNames <- case_match(ri$CITY, "CARY"~"CARY", "RALEIGH"~"RALEIGH",
 ↪  "KNIGHTDALE"~"KNIGHTDALE", "CLAYTON"~"CLAYTON", "FUQUAY
 ↪  VARINA"~"FUQUAY-VARINA", "FUQUAY-VARINA"~"FUQUAY-VARINA",
                           "GARNER"~"GARNER", "MORRISVILLE"~"MORRISVILLE",
                            ↪  "MORRISVILE"~"MORRISVILLE", "RESEARCH TRIANGLE
                            ↪  PARK"~"RTP", "RTP"~"RTP",
                           "WENDELL"~"WENDELL", "APEX"~"APEX", "WILLOW
                            ↪  SPRING"~"WILLOW SPRING", "HOLLY
                            ↪  SPRINGS"~"HOLLY SPRINGS", "HOLLY
                            ↪  SPRING"~"HOLLY SPRINGS",
                           "ROLESVILLE"~"ROLESVILLE", "ZEBULON"~"ZEBULON",
                            ↪  "WAKE FOREST"~"WAKE FOREST", "NEW HILL"~"NEW
                            ↪  HILL", "ANGIER"~"ANGIER", "NORTH
                            ↪  CAROLINA"~"NORTH CAROLINA")
#used the case match command from the sfpark exercise. wow that sucked I hope
 ↪  there's a better way to do it. I noticed that if I did only the ones that
 ↪  needed changing it would convert all others to NA so it's like you have
 ↪  to type out each one to preserve it.

scorebycity <- ri %>%
  group_by(CityNames) %>%
  summarize(meanscore = mean(SCORE))
#created new dataframe to average the inspection scores for each city. There
 ↪  are two that don't make much sense, "north carolina" and NA

ggplot(data=scorebycity, aes(x=CityNames, y=meanscore)) + geom_point() +
 ↪  theme(axis.title = element_text(size=10),
        axis.text = element_text(size=6),
        axis.text.x = element_text(angle=90))
```



```
#There are differences between cities. I cannot find a way to order the
↪  scatterplot in ascending or descending order of scores, but it would be
↪  visually helpful. The top 3 are New Hill, Research Triangle Park, and
↪  Holly Springs, the only 3 to average a score above 98. Angier and Zebulon
↪  are the lowest, scoring below 95.
```

Question 4

```
scorebyinspector <- ri %>%
  group_by(INSPECTOR) %>%
  summarize(meanscore = mean(SCORE))
#made new dataframe to average the inspection scores for each inspector

ggplot(data=scorebyinspector, aes(x=INSPECTOR, y=meanscore)) + geom_point() +
↪  theme(axis.title = element_text(size=10),
        axis.text = element_text(size=6),
        axis.text.x = element_text(angle=90))
```



```
#Thomas Jumalon scores much lower than the rest
```

Question 5

```
ri$SCORE[ri$INSPECTOR=="Thomas Jumalon"] #Thomas has only 3 recorded
 ↪  inspections of 91, 91, and 85.
```

```
[1] 91 91 85
```

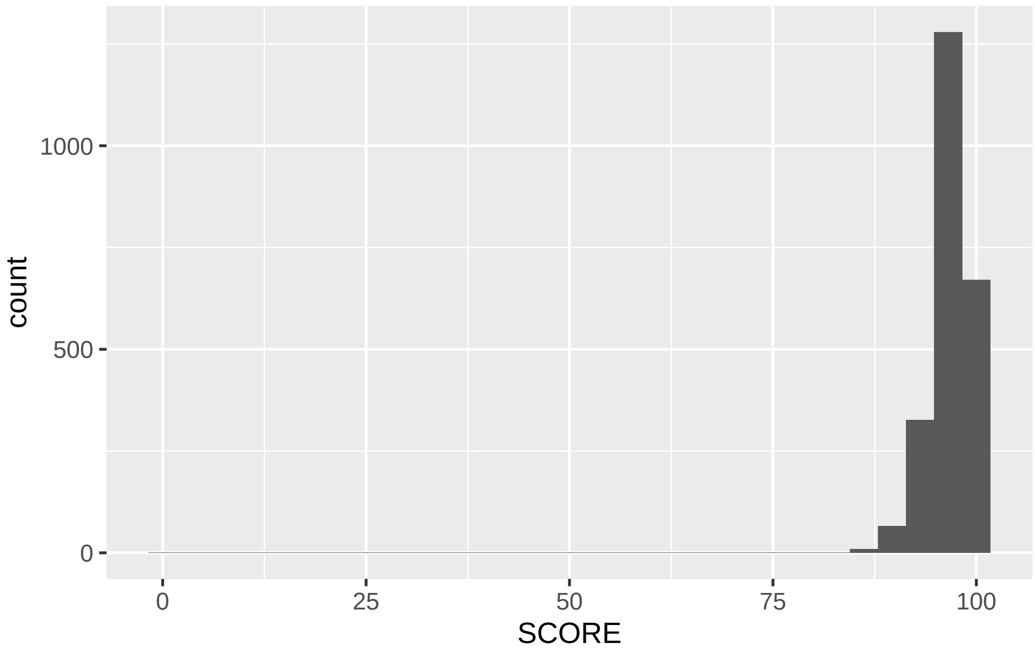
```
sampleinspector <- ri %>%
  group_by(INSPECTOR) %>%
  summarize(totalentries = length(SCORE))
#Many of the inspectors have single digit inspections logged, but Mr. Jumalon
 ↪  is the only one that stands out to an extreme degree in the scores given.
 ↪

samplecity <- ri %>%
  group_by(CityNames) %>%
  summarize(totalentries = length(SCORE))
#Angier, New Hill, Willow Spring, Clayton, and RTP all have very low sample
 ↪  sizes, each less than 4. This could explain Angier's low score, with only
 ↪  1 inspection recorded. Conversely, New Hill and RTP are at the top each
 ↪  with only 2 inspections recorded.
```

Question 6

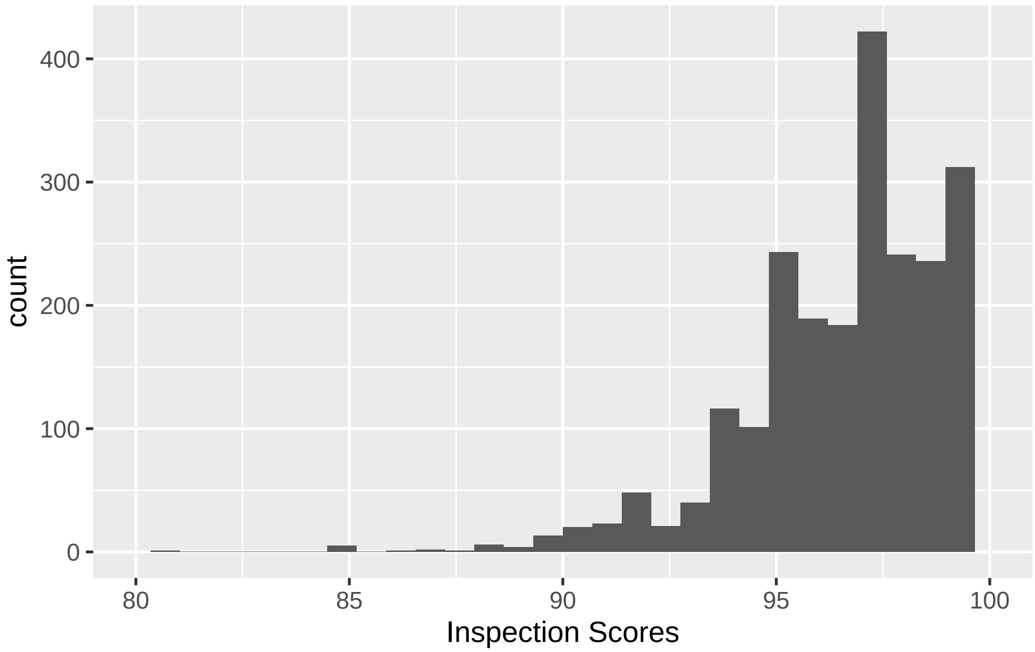
```
scorebyfacility <- ri %>%
  group_by(FACILITYTYPE) %>%
  summarize(meanscore=mean(SCORE))
#Restaurants score lowest out of all categories. Yikes
```

Question 7

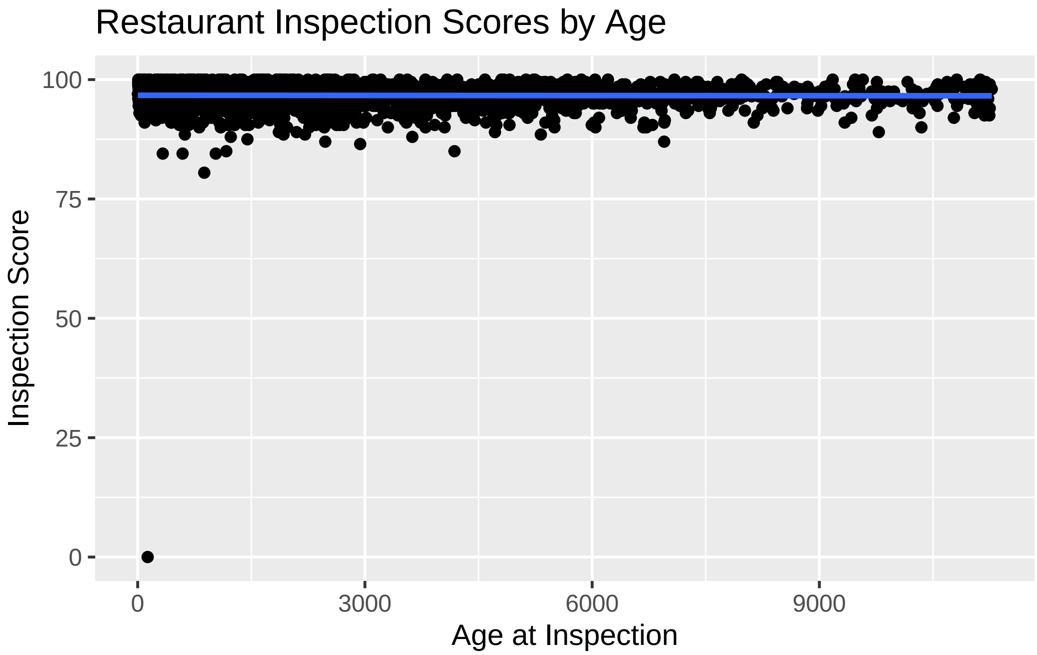
```
restonly <- ri %>%
  filter(FACILITYTYPE=="Restaurant")
#Created a dataset excluding any facility other than restaurants to do
 ↪  analysis on restaurants only

ggplot(data=restonly, aes(x=SCORE)) + geom_histogram()
```



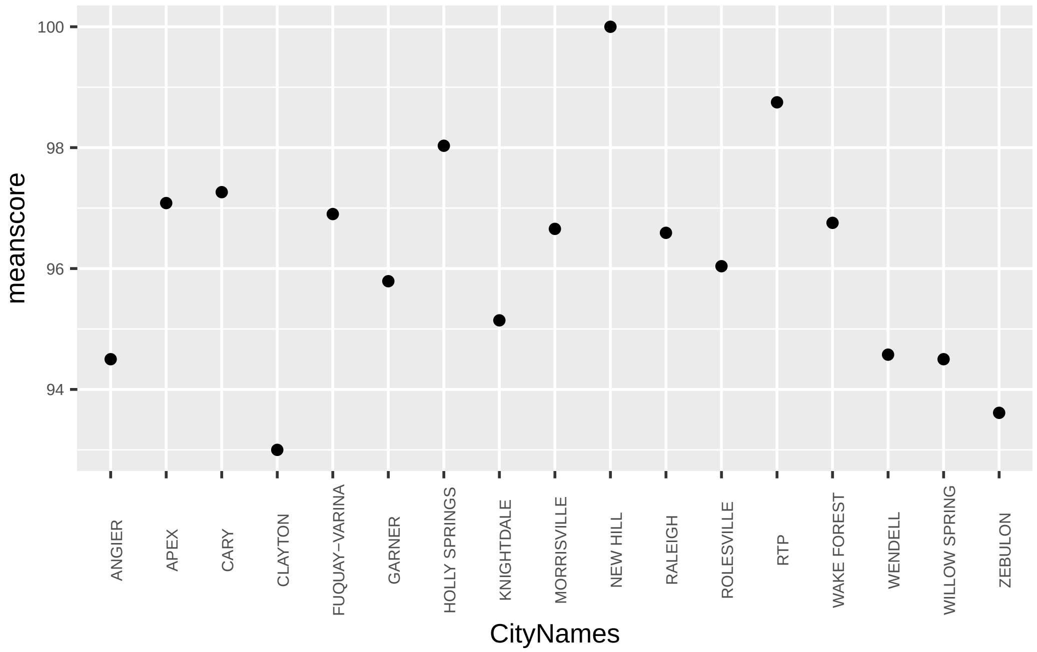
```
#there is a single "0" score, but would be useful to set xlim to see a more
↪  visually instructive distribution
ggplot(data=restonly, aes(x=SCORE)) + geom_histogram() + xlim(80,100) +
↪  xlab("Inspection Scores")
```



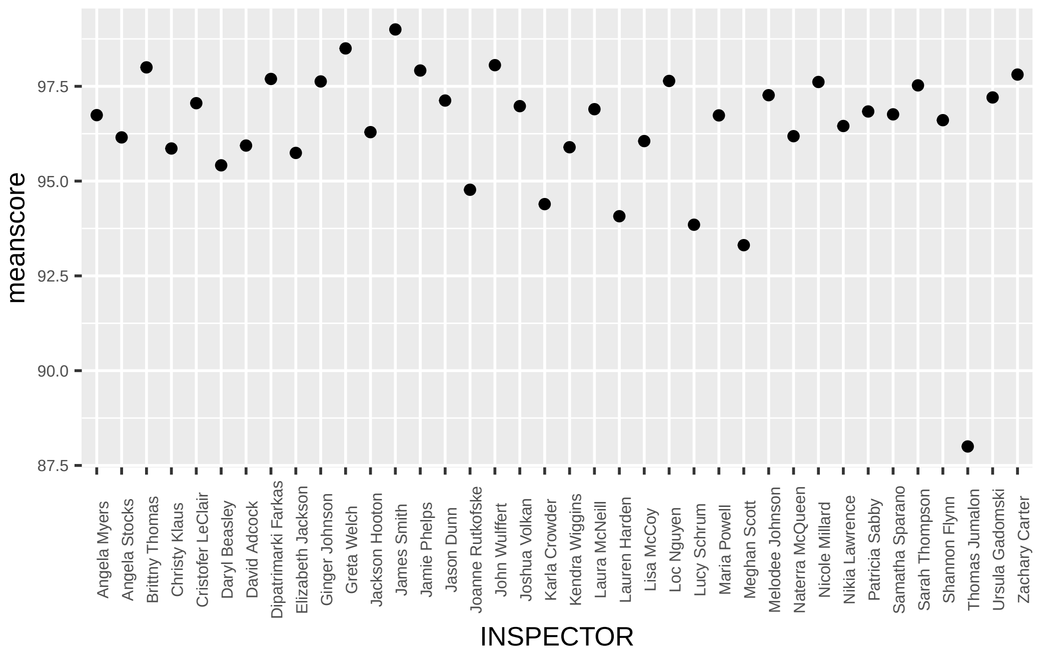
```
#visualized inspection scores with histogram


fit3 <- lm(SCORE~age_to_inspection, data=restonly)
ggplot(data=fit3, aes(x=age_to_inspection, y=SCORE)) + geom_point() +
↪  geom_smooth(method="lm") + labs(x="Age at Inspection", y="Inspection
↪  Score", title="Restaurant Inspection Scores by Age")
```



```
#No discernable relationship, there are simply more young restaurants than
 ↪  old ones


restbycity <- restonly %>%
  group_by(CityNames) %>%
  summarize(meanscore = mean(SCORE))
#fairly significant difference in scores, but we already know that several of
 ↪  those are due to low sample size

ggplot(data=restbycity, aes(x=CityNames, y=meanscore)) + geom_point() +
 ↪  theme(axis.title = element_text(size=10),
        axis.text = element_text(size=6),
        axis.text.x = element_text(angle=90))
```



```
#plot of restaurant scores by city

restbyinspector <- restonly %>%
  group_by(INSPECTOR) %>%
  summarize(meanscore = mean(SCORE))
#Thomas must've seen some nasty stuff.

ggplot(data=restbyinspector, aes(x=INSPECTOR, y=meanscore)) + geom_point() +
↪  theme(axis.title = element_text(size=10),
        axis.text = element_text(size=6),
        axis.text.x = element_text(angle=90))
```



```
#Data similar, scores are a bit lower. To be expected after seeing that
 ↪  restaurants as a category score the lowest.

samplerestinspector <- restonly %>%
  group_by(INSPECTOR) %>%
  summarize(totalentries = length(SCORE))
#Six with single digit inspections recorded. James Smith, Greta Welch,
 ↪  Brittany Thomas, three of the top four by rating and also the three least
 ↪  voluminous in sample size.

samplerestcity <- restonly %>%
  group_by(CityNames) %>%
  summarize(totalentries = length(SCORE))
#Many of the highest and lowest scores are the lowest sample size
```