NYC Urban Park Ranger Rescues

This year marks 40 years since New York City's Urban Park Rangers were first enlisted under Mayor Koch in 1979 ("Urban Park Rangers: NYC Parks," n.d.-a). They were part of the plan to revitalize the city's parks that had suffered years of neglect and vandalism. The public response was enthusiastic and continues to be so ("How a Smokey Bear hat inspired the Urban Park Rangers," n.d.). The NYC Department of Parks and Recreation is celebrating with new and expanded programs for city residents. Urban Rangers are not only custodians of the parks but the public face of green spaces across the five boroughs (*Urban Park Rangers: NYC Parks*, n.d.-b)g}. Their varied duties include education, conservation, and wildlife rescue. The latter being the subject of this dataset ("Urban Park Ranger Animal Condition Response | NYC Open Data," 2018) obtained through the NYC Open Data site.

NYC Open Data provides the public with datasets from the city's various agencies and partners in an effort to provide not only transparency but to invite innovation ("NYC Open Data," n.d.). The Department of Parks and Recreation offers over one hundred datasets to the public including one for Urban Park Ranger Animal Condition Response. This dataset covers incoming requests for assistance with animals in distress within city parks completed by Urban Park Rangers. First posted on October 31, 2018, the dataset has 982 entries as of June 27, 2019. The collection is updated biannually and has 22 variables.

The data available covers location, time, call source, descriptive information, interaction with other agencies, and outcome. The dataset had some missing fields are due to the fact that the information wasn't applicable to that request. If the call was not made through 311, there was 311 Service Request Number. It is easy to guess which fields in the dataset were inputted manually due to inconsistencies. One issue with the dataset was the inconsistent spelling and letter-case of individual species. A raccoon could be "Raccoon" or "RACCOON". The site of a request was recorded under the Property variable (park or recreational area name) and then included a more descriptive variable for location within the park. The Location variable included a description (i.e. near the pond) within a property but sometimes had GPS coordinates or N/A. The N/A could be the result of the call being unfounded. When the Final Ranger Action variable is "Unfounded", it is unclear if this is due to the animal not being present or if the situation was not a concern. The time of the initial call and the time of response were also in two different formats.

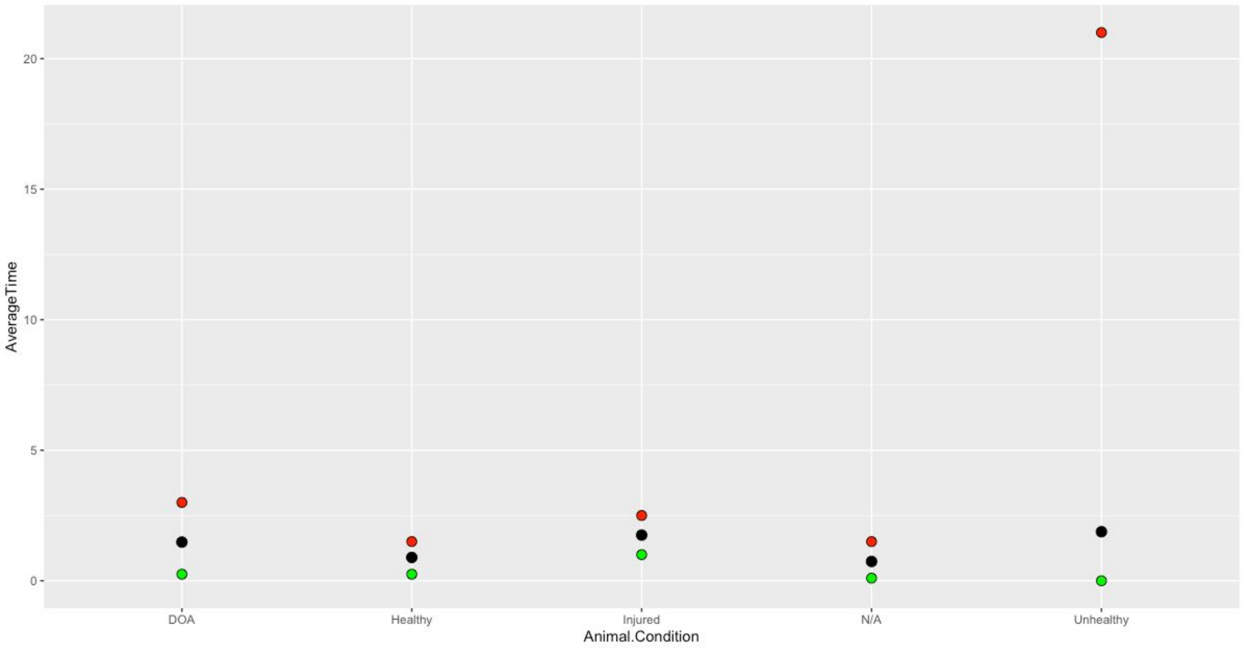
Although this dataset has a good number of entries, the time period it covers is less than a year. It would be better to have more time to determine if the number of calls for a certain species is increasing or if types of animal requests are seasonal.

Another challenge with this dataset was the date and time value for initial call and arrival. Although it was consistent, it was not a format recognized by R. Using the Lubridate library, the time difference between the time of call and response was able to be calculated. After doing this, it was possible to do calculations on average response times.

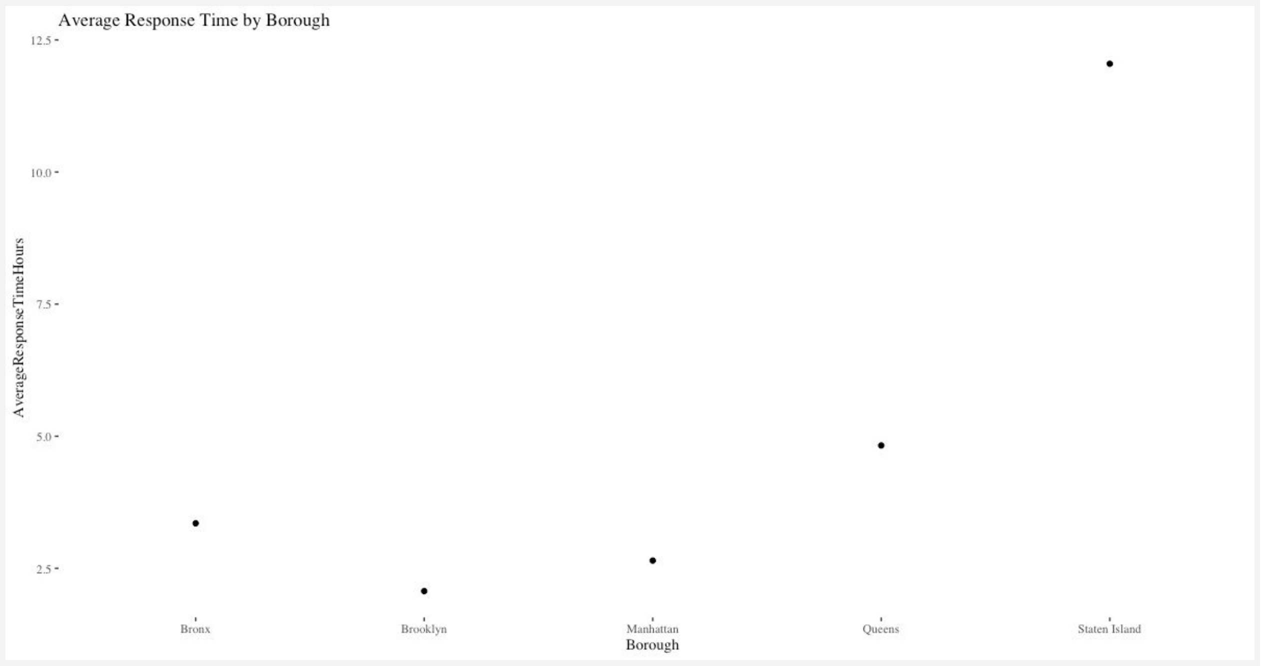
Although Rangers cover all five boroughs, Manhattan, and notably, Central Park is the top location. Prospect Park in Brooklyn is another large park and comes in second. The calls come from varied sources including 311 but most come from public-interest conservation groups. The top culprit overall is the raccoon, but NYC is also native to the Red-Tailed Hawk which also ranks high on the list. The majority of animals end up in the Animal Control Center. Some are rehabilitated, relocated and/or simply monitored.

Small mammals on the rabies vector scale (RVS) is the most frequent animal class and includes opossums and raccoons. You could infer that raccoons are incredibly pervasive in the city from the fact they top the chart in number of calls. They are a concern due to the fact that they can carry rabies and this could have an effect on the amount as well. If this data was available ten years ago, we could see if this percentage has changed.

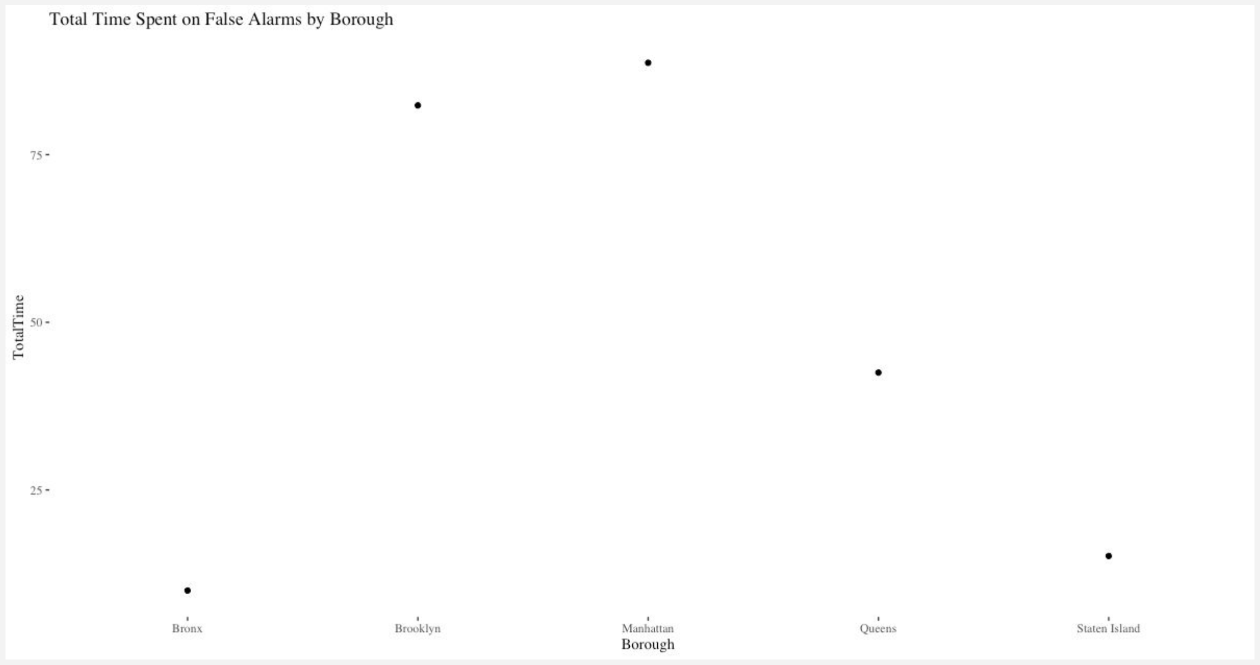
In most calls throughout the city, the reason is an unhealthy adult animal. When we look at calls for raccoons in Prospect Park, the same is true. The average duration of response of all calls is one hour, but for raccoons, it is 1.5. The chart below shows the average time for the duration of response for Prospect raccoons in black broken out into the condition of the animal. We can assume that when the condition is N/A, the ranger couldn't get close enough or it couldn't be found. The maximum time is in red and the minimum is in red. One unhealthy raccoon got 21 hours of attention from the rangers.



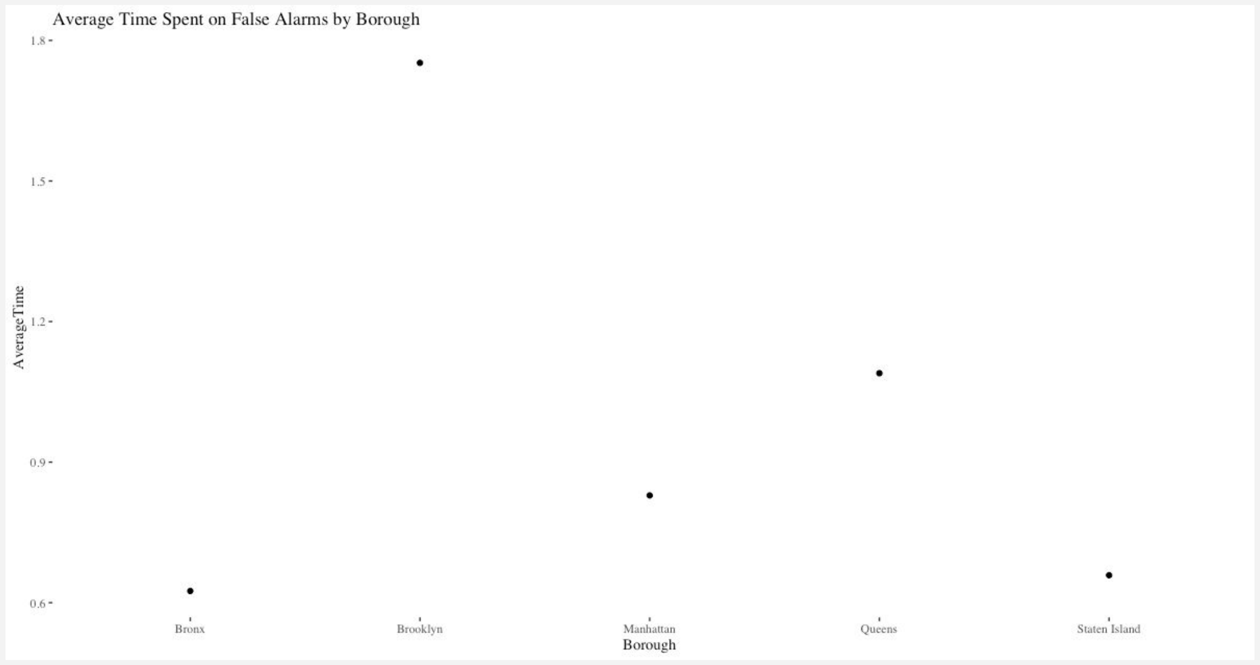
From the time a call first comes in until the moment an urban ranger arrives varies greatly. Average response time appears to be the best in Brooklyn. Staten Island has the longest by far with the average being over twelve hours.



There are a fair number of calls that are determined to be unfounded. When total time is calculated for each borough, we see that Manhattan and Brooklyn have a much larger amount. This makes sense due to the fact they have a larger volume of calls in general and two large popular parks. The average time spent on these calls does not follow the same order. Brooklyn durations are much higher than the next longest, which is Queens. Manhattan comes in third although they have the highest in total time.



Time is shown in hours (above and below).



When most people think of New York City, they may imagine squirrels or pigeons in the park. Many may be surprised to learn that our city has a large and varied population of native wildlife, from raptors to raccoons, and even coyotes in the Bronx ("Urban Wildlife: Urban Park Rangers: NYC Parks," n.d.). The records also included a call for a harbor seal and a dolphin. It will be interesting to see how what else shows up in the NYC Urban Ranger database over time.

Bibliography

How a Smokey Bear hat inspired the Urban Park Rangers. (n.d.). Retrieved September 23, 2019, from Am New York website: https://www.amny.com/news/nyc-urban-park-rangers-1.30443288

NYC Open Data. (n.d.). Retrieved September 22, 2019, from NYC Open Data website: https://opendata.cityofnewyork.us/

Urban Park Ranger Animal Condition Response | NYC Open Data. (2018, October 31). Retrieved September 21, 2019, from NYC Open Data website: https://data.cityofnewyork.us/Environment/Urban-Park-Ranger-Animal-Condition-Response/fuhs-xmg2

Urban Park Rangers: NYC Parks. (n.d.-a). Retrieved September 22, 2019, from https://www.nycgovparks.org/events/urbanparkrangers

Urban Park Rangers: NYC Parks. (n.d.-b). Retrieved from https://www.nycgovparks.org/programs/rangers

Urban Wildlife: Urban Park Rangers: NYC Parks. (n.d.). Retrieved September 23, 2019, from https://www.nycgovparks.org/programs/rangers/wildlife-management

Code in R

```
# Descriptive Data Analysis

# Urban Park Ranger Animal Condition Response from Open Data


library(tidyverse)

library(dplyr)

library(lubridate)

library(ggthemes)


# Specified that the first row is the header, and check the formatting of names

rangerRescues <- read.csv("../DataSets/Urban_Park_Ranger_Animal_Condition_Response.csv", header=TRUE, check.names = TRUE)


# Shows that it is being read as a data frame

class(rangerRescues)

# The dataframe has 982 objects with 22 variables

dim(rangerRescues)

# List all the variable names

names(rangerRescues)

# Overview of dataset shows that date and time are being read as a factor

str(rangerRescues)
```

```
# A better overview...

glimpse(rangerRescues)


# Statistical summary

# The larger parks see the most wildlife requests. Manhattan's Central Park and Brooklyn's Prospect Park get the majority of the calls.

# Most calls are in Manhattan, are made by conservancies (animal interest groups), in parks, mostly Central Park, and are about small mammals, particularly raccoons.

# Rarely are the police involved. The longest time spent on response was 21 hours but most calls seem to take around an hour.

# The majority of calls are for Small Mammals on the Rabies Vector Scale and Raccoons make up nearly all of these requests.

summary(rangerRescues)


# Check beginning and end

head(rangerRescues)

# Starts with a hawk, goose, parrot, chicken and a Red-Eared Slider, which happens to be an invasive reptile

tail(rangerRescues)

# Ends with a hawk and a group of raccoons.


sum(is.na(rangerRescues))

# There are 870 missing values in the dataset. Why? Some fields are not always applicable.
# Create new dataset to manipulate

rangerRescues1 <- rangerRescues[,]

summary(rangerRescues1)


# Convert to characters

rangerRescues1$Date.and.Time.of.initial.call <-
as.character(rangerRescues1$Date.and.Time.of.initial.call)

rangerRescues1$Date.and.time.of.Ranger.response <-
as.character(rangerRescues1$Date.and.time.of.Ranger.response)

rangerRescues1$Species.Description <- as.character(rangerRescues1$Species.Description)


# Fix inconsistencies in species case

rangerRescues1$Species.Description <- tolower(rangerRescues1$Species.Description)

# Check it

table(rangerRescues1$Species.Description)

head(subset(rangerRescues1, select = 'Species.Description'))

# rd-tailed hawk, red tailed hawk, red tail, red-tailed hawk (correct), red tailed-hawk

# racoon, raccoon (correct)

# deer, white tailed deer, white-tailed deer (correct)

# unknown (correct), unknown species,n/a

# rabbit (correct), pet rabbit
# Wildcard selection in R?!!
?grep
rangerRescues1$Species.Description[rangerRescues1$Species.Description=="red tailed hawk"] <- "red-tailed hawk"
rangerRescues1$Species.Description[rangerRescues1$Species.Description=="racoon"] <- "raccoon"
rangerRescues1$Species.Description[rangerRescues1$Species.Description=="deer"] <- "white-tailed deer"
rangerRescues1$Species.Description[rangerRescues1$Species.Description=="unknown species"] <- "unknown"
rangerRescues1$Species.Description[rangerRescues1$Species.Description=="pet rabbit"] <- "rabbit"
rangerRescues1$Species.Description[rangerRescues1$Species.Description=="turtle/ unspecified species"] <- "turtle"


# Animal.Class
table(rangerRescues1$Animal.Class)


# Find typical response time - need to convert data
# Format Dates | Test one value
x <- "06/10/2019 01:00:00 PM"
parse_date_time(x, '%m/%d/%Y %I:%M:%S %p', tz = "EST")
x <- parse_date_time(x, '%m/%d/%Y %I:%M:%S %p', tz = "EST")
x
```

```
# Convert to Dates

rangerRescues1$Date.and.Time.of.initial.call <- parse_date_time(rangerRescues1$Date.and.Time.of.initial.call, '%m/%d/%Y %I:%M:%S %p', tz = "EST")

rangerRescues1$Date.and.time.of.Ranger.response <- parse_date_time(rangerRescues1$Date.and.time.of.Ranger.response, '%m/%d/%Y %I:%M:%S %p', tz = "EST")

head(rangerRescues1)



# Create new dataset with times and a new column with response time (borough)

?dplyr

rescuesResponse <- select(rangerRescues1, Date.and.Time.of.initial.call, Date.and.time.of.Ranger.response, Borough)

head(rescuesResponse)



# create by Borough dataframe

rescuesResponseTime_byBorough <- rescuesResponse %>%

  group_by (Borough) %>%

  summarize(AverageResponseTimeHours =(mean(Date.and.time.of.Ranger.response - Date.and.Time.of.initial.call)/3600) )



# create plot chart

ggplot(rescuesResponseTime_byBorough, aes(x = Borough, y = AverageResponseTimeHours)) +
  geom_point() +
  labs(title = "Average Response Time by Borough") +
  theme_tufte()


# Create set just for Manhattan
rescuesManhattan <- rangerRescues1 %>% filter(Borough=="Manhattan")
head(rescuesManhattan, 10)


# Create set just for Raccoons in Prospect Park, Brooklyn
rescuesProspectRaccoons <- rangerRescues1 %>% filter(Borough=="Brooklyn", Property=="Prospect Park", Species.Description=="raccoon")
summary(rescuesProspectRaccoons)
rescuesProspectRaccoons <- select(rescuesProspectRaccoons, -Borough, -Property, -Species.Description, -Species.Status, -Animal.Class)
head(rescuesProspectRaccoons)


# Change empty Animal.Condition fields to N/A
rescuesProspectRaccoons$Animal.Condition[rescuesProspectRaccoons$Animal.Condition==""] <- "N/A"
write.csv(rescuesProspectRaccoons, "../DataSets/rescuesProspectRaccoons.csv")


# Arrange by Condition
```

```
rescuesProspectRaccoons %>% arrange(Animal.Condition)


# View duration of responses
table(rescuesProspectRaccoons$Animal.Condition, rescuesProspectRaccoons$Duration.of.Response)
# Raccoons that are unhealthy usually take 1-2 hours, most unhealthy calls are 2 hours


# Average duration of response by condition
rescuesProspectRaccoonsDuration <- rescuesProspectRaccoons %>%
  group_by(Animal.Condition) %>%
  summarize(AverageTime = mean(Duration.of.Response), MaxTime = max(Duration.of.Response), MinTime = min(Duration.of.Response))
rescuesProspectRaccoonsDuration


# Plot average duration (min, max) of response by condition
ggplot(rescuesProspectRaccoonsDuration, aes(x = Animal.Condition, y = AverageTime)) +
  geom_point(shape=20, size=5, fill="#00000080")+
  geom_point(data=rescuesProspectRaccoonsDuration, aes(y=MaxTime),shape=21, size=3, fill="#FF0000FF") +
  geom_point(data=rescuesProspectRaccoonsDuration, aes(y=MinTime),shape=21, size=3, fill="#00FF00FF")
  #labs(title = "Prospect Park Racoons Response Average Duration with Max and Min")
 # Outlier for 21 hours for an unhealthy racoon.
# Frequency by animal - find most common and outliers

rescuesAnimal <- select(rangerRescues1, "Species.Description", "Species.Status")

head(rescuesAnimal)


# Frequency by age of animal by Borough

table(rangerRescues1$Age, rangerRescues1$Borough)


# Frequency by borough and final action

table(rangerRescues$Final.Ranger.Action, rangerRescues1$Borough)

rescuesBoroughs <- as.data.frame (table(rangerRescues$Final.Ranger.Action, rangerRescues1$Borough))

glimpse(rescuesBoroughs)


# Frequency by Animal Condition, Rehabilitation

table(rangerRescues1$Animal.Condition) # The largest reported group was unhealthy. 106

table(rangerRescues1$Rehabilitator) #The overwhelming majority do not go to a rehabilitation center


# False Alarms

falseAlarms <- rangerRescues1 %>% filter(Final.Ranger.Action=="Unfounded")
```

```
glimpse(falseAlarms)
# Average time of False Alarm Call by Borough
falseAlarms %>%
  group_by(Borough) %>%
  summarize(AverageTime = mean(Duration.of.Response))


# create by Borough dataframe
falseAlarms_byBorough <- falseAlarms %>%
  group_by (Borough) %>%
  summarize(AverageTime = mean(Duration.of.Response), TotalTime=sum(Duration.of.Response))


# create plot chart
ggplot(falseAlarms_byBorough, aes(x = Borough, y = AverageTime)) +
  geom_point() +
  labs(title = "Average Time Spent on False Alarms by Borough") +
  theme_tufte()
# Brooklyn unfounded calls take the longest on average


ggplot(falseAlarms_byBorough, aes(x = Borough, y = TotalTime)) +
  geom_point() +
  #geom_hline(data=falseAlarms_byBorough, aes(yintercept=AverageTime)) #not worth looking at together

  labs(title = "Total Time Spent on False Alarms by Borough") +

  theme_tufte()

# Manhattan spends the most total time on unfounded calls and this makes sense because it has the highest call volume
```