Data Hackathon Visualization

Ana Hernandez, Adria Vazquez, Nataniel Tsai, Ailssa Villa, Ycied Talavera

7/29/2020

Data Visualization

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ---------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

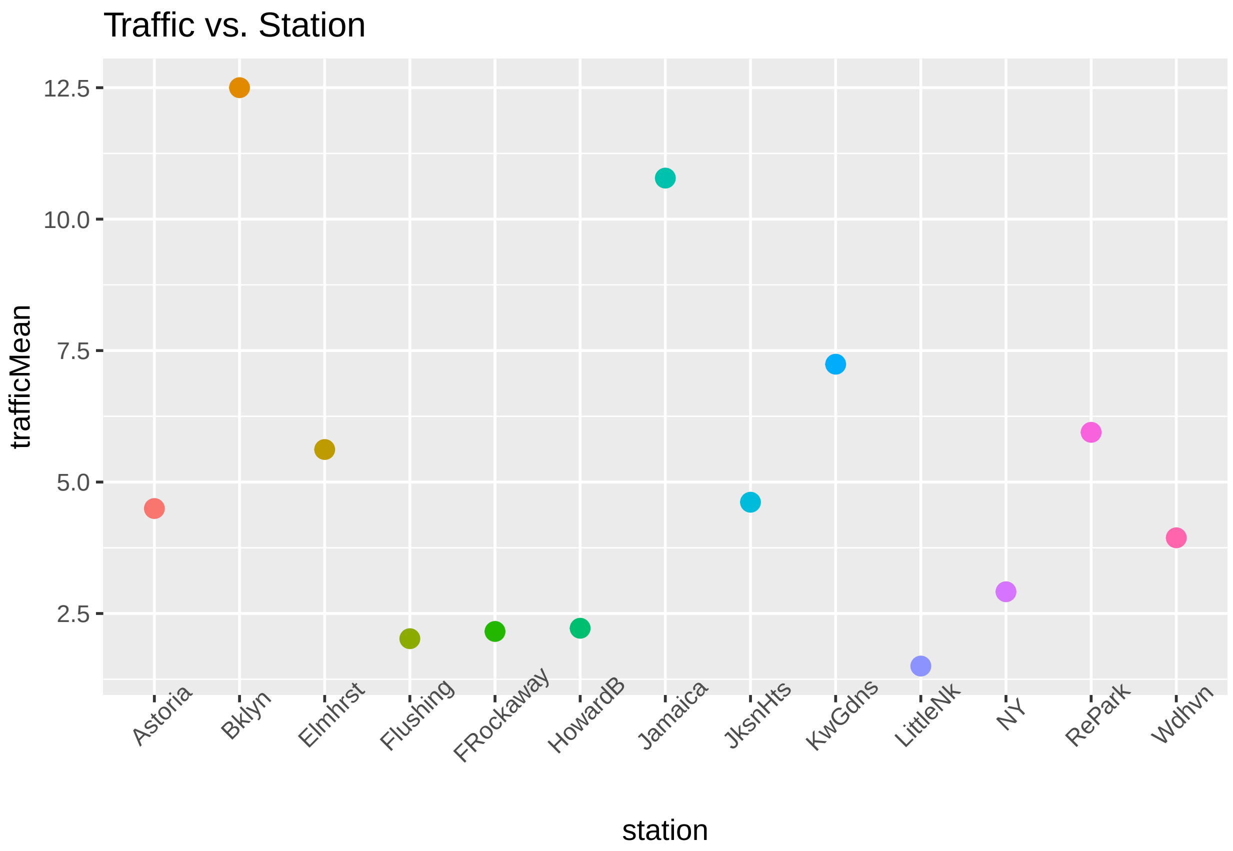
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
data <- read_excel("072020NYCMTAJuly2014.xlsx")

station <- names(data)

price <- c(2.750, 2.500, 2.500, 2.456, 2.435, 2.39, 2.301,
           2.281, 2.250, 2.136, 2.131, 1.950, 1.917)
trafficMean <- rep(-1, length(data))

trafficMean[1] <- mean(na.exclude(data$HowardB))/1000
trafficMean[2] = mean(na.exclude(data$LittleNk))/1000
trafficMean[3] = mean(na.exclude(data$FRockaway))/1000
trafficMean[4] = mean(na.exclude(data$NY))/1000
trafficMean[5] = mean(na.exclude(data$Flushing))/1000
```

```
trafficMean[6] = mean(na.exclude(data$RePark))/1000
trafficMean[7] = mean(na.exclude(data$Bklyn))/1000
trafficMean[8] = mean(na.exclude(data$Astoria))/1000
trafficMean[9] = mean(na.exclude(data$KwGdns))/1000
trafficMean[10] = mean(na.exclude(data$Jamaica))/1000
trafficMean[11] = mean(na.exclude(data$JksnHts))/1000
trafficMean[12] = mean(na.exclude(data$Elmhrst))/1000
trafficMean[13] = mean(na.exclude(data$Wdhvn))/1000

tbl <- tibble(station, price, trafficMean)
```

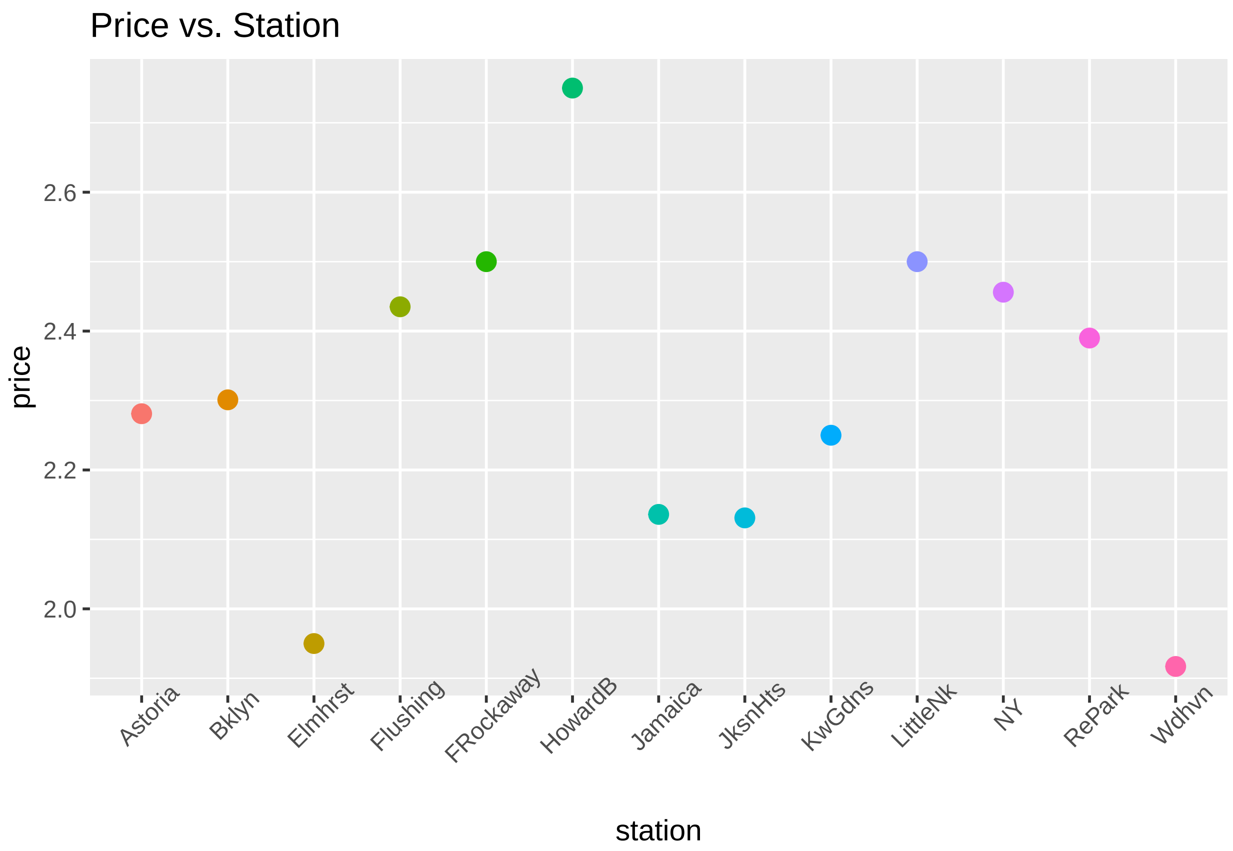
Graph of traffic means

```
ggplot(tbl, aes(x = station, y = trafficMean)) +
  geom_point(aes(color = station), size = 3) +
  scale_shape_manual(values = seq(0,13)) +
  labs(title = "Traffic vs. Station") +
  theme(axis.text.x = element_text(angle = 45), legend.position = "none")
```



Graph of price means

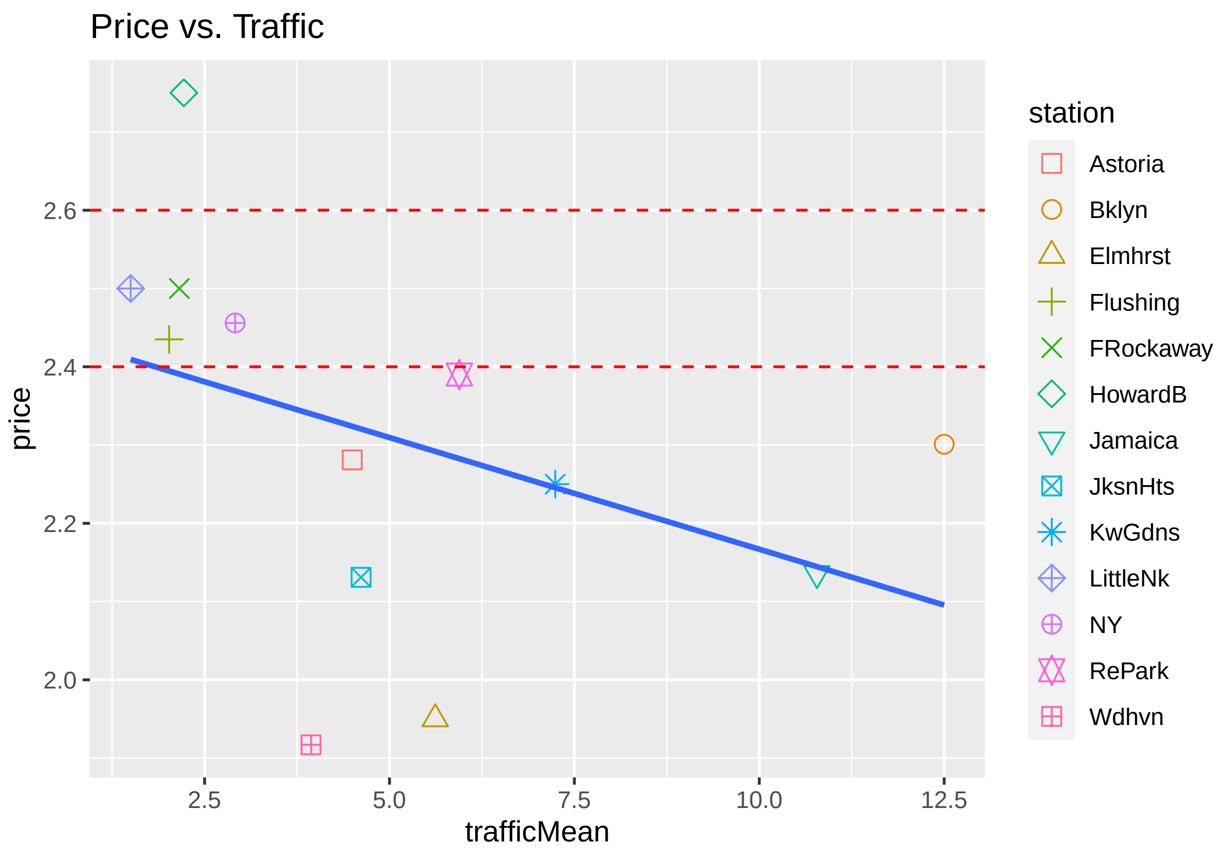
```
ggplot(tbl, aes(x = station, y = price)) +
  geom_point(aes(color = station), size = 3) +
  scale_shape_manual(values = seq(0,13)) +
  labs(title = "Price vs. Station") +
  theme(axis.text.x = element_text(angle = 45), legend.position = "none")
```



Graph of Traffic vs. Price means

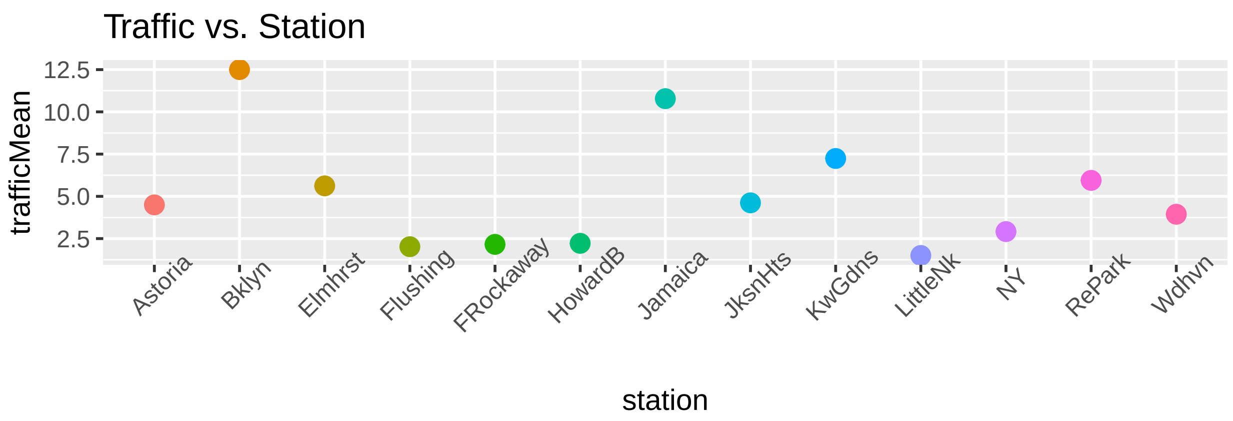
```
ggplot(tbl, aes(x = trafficMean, y = price)) +
  geom_point(aes(color = station, shape = station), size = 3) +
  scale_shape_manual(values=seq(0,13)) +
  labs(title = "Price vs. Traffic") +
  stat_smooth(method = "lm", se = FALSE, fullrange = TRUE) +
  geom_hline(yintercept = 2.4, linetype = "dashed", color = "red") +
  geom_hline(yintercept = 2.6, linetype = "dashed", color = "red")
```

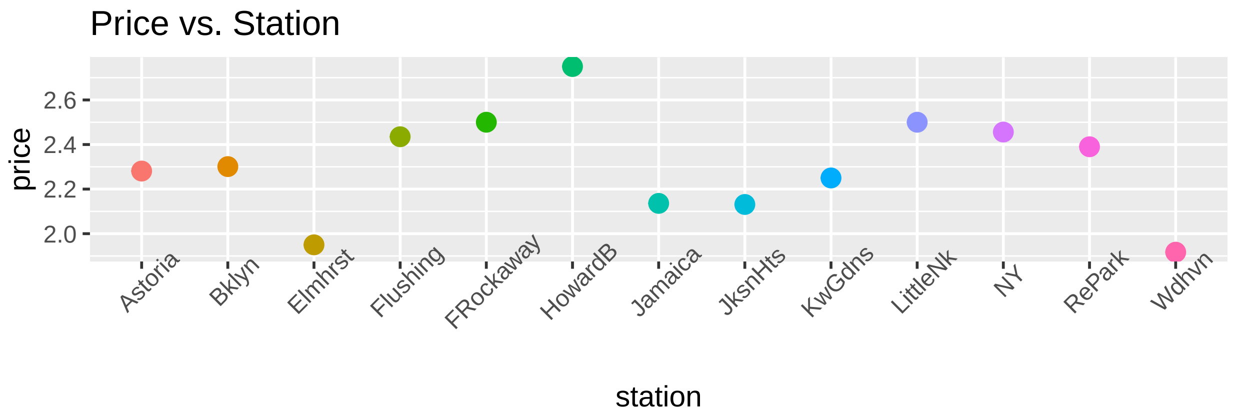
```
## `geom_smooth()` using formula 'y ~ x'
```



Comparing both mean graphs

```
# Plot of traffic mean with stations at the bottom
p1 <- ggplot(tbl, aes(x = station, y = trafficMean)) +
  geom_point(aes(color = station), size = 3) +
  scale_shape_manual(values = seq(0,13)) +
  labs(title = "Traffic vs. Station") +
  theme(axis.text.x = element_text(angle = 45), legend.position = "none")

# Plot of price
p2 <- ggplot(tbl, aes(x = station, y = price)) +
  geom_point(aes(color = station), size = 3) +
  scale_shape_manual(values = seq(0,13)) +
  labs(title = "Price vs. Station") +
  theme(axis.text.x = element_text(angle = 45), legend.position = "none")

grid.arrange(p1, p2, ncol = 1, nrow = 2)
```





Summary

```
fit <- lm(price ~ trafficMean)
summary(fit)
```

```
##
## Call:
## lm(formula = price ~ trafficMean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42280 -0.04291  0.04045  0.10742  0.36111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.45222    0.11425  21.464  2.5e-10 ***
## trafficMean -0.02853    0.01893  -1.507     0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.223 on 11 degrees of freedom
## Multiple R-squared:  0.1711, Adjusted R-squared:  0.09576
## F-statistic: 2.271 on 1 and 11 DF,  p-value: 0.16
```