Lab 1: Fremont Bridge Bicycle Data

June 22, 2016

If you've ever walked or biked across the Fremont bridge, you might have seen the bicycle counter which keeps track of how many bikes cross the bridge. We will be analyzing data from that counter which is available in raw form at `http://data.seattle.gov`. Today, we will be analyzing data from that machine recorded earlier this year.

Figure 1: Image via `http://www.bicycleretailer.com/`

Loading the data

First, let's load the data

```
# If you want to access the file directly from my site
bike.crossings <- read.csv("http://www.stat.washington.edu/~ysamwang/notes/bike_crossings.csv")

# If you download and save the file locally
bike.crossings <- read.csv("bike_crossings.csv")
```

Let's take a look at what's in the data. We can use the **head** function to view the first few lines of our data.

```
head(bike.crossings)

##       Date Hour WestSide EastSide Day_of_week_name Day Month_name Weekend
## 1 1/1/2016    0       18        9              Fri   1        Jan       0
## 2 1/1/2016    1       15        3              Fri   1        Jan       0
## 3 1/1/2016    2       11        6              Fri   1        Jan       0
## 4 1/1/2016    3        7        1              Fri   1        Jan       0
## 5 1/1/2016    4        2        0              Fri   1        Jan       0
## 6 1/1/2016    5        6        4              Fri   1        Jan       0
```

We can see the following variables-

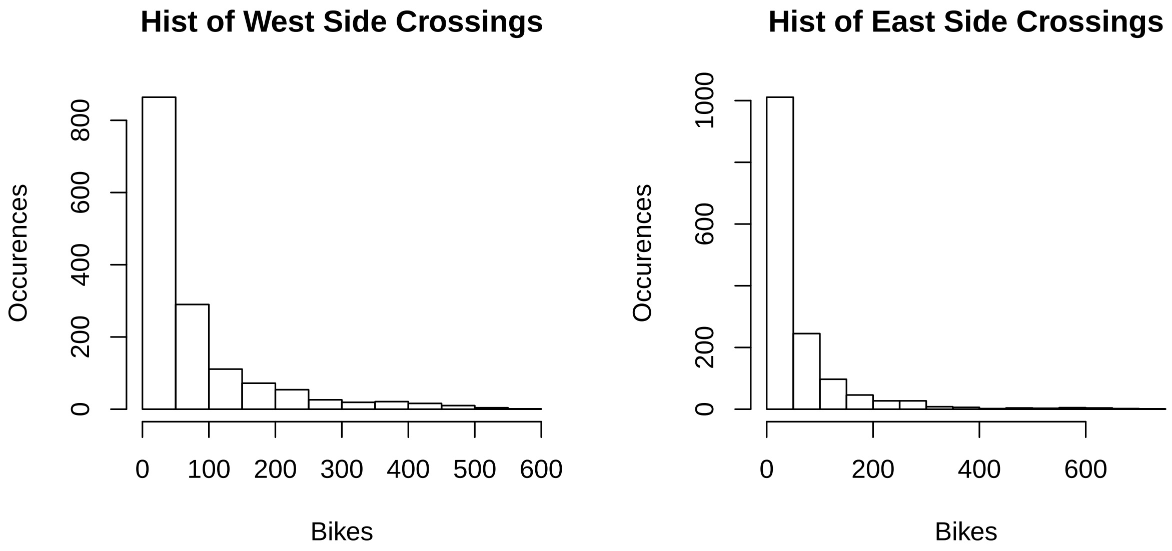
- Date: Day, month and year of measurement
- Hour: Time of Day in 24-hr clock (ie 0 is midnight, 12 is noon)
- WestSide: Number of bikes crossed during the hour on the west side of Fremont Bridge
- EastSide: Number of bikes crossed during the hour on the east side of Fremont Bridge
- Day_of_week_name: Day of the week
- Day: Day of the month
- Month_name: Name of month
- Weekend: 0 if weekday, 1 if weekend

We can also use the `dim` command to get the dimensions of the data

```
# 1488 rows and 8 columns
dim(bike.crossings)

## [1] 1488    8
```

When there is a table with multiple columns, we can use the `$` operator to pull out specific columns. For example `bike.crossings$WestSide` will return the `WestSide` column from `bike.crossings`. Notice for the `hist` command, we include the following arguments to label the plot (`main` is the main title, `ylab` is the label for the y-axis and `xlab` is the label for the x-axis). We can first view a histogram of the number of crossings which tells us how many times a specific number occured in our data set. For instance, we can see for the west side of the bridge, there were over 800 hour time periods, where between 0-50 bikes crossed the bridge.

```
hist(bike.crossings$WestSide, main = "Hist of West Side Crossings",
     ylab = "Occurences", xlab = "Bikes")
```



Questions

- What type of variable is `WestSide`? Categorical or numerical?

- What type of variable is `Month_name`? Categorical or numerical?
- Describe the shape of the distribution.

Subsetting the data

Is there a difference in the amount of traffic during January and May? Let's take a look. First, note how we can grab specific elements of a vector using the the square brackets.

```
# Get the first element
# Note that the first element is index
# by 1 (instead of 0 as is common in other languages)
bike.crossings$WestSide[1]

## [1] 18

# Get the first 5 elements
bike.crossings$WestSide[c(1, 2, 3, 4, 5)]

## [1] 18 15 11  7  2

bike.crossings$WestSide[1:5]

## [1] 18 15 11  7  2

# Get the first row of bike.crossings
bike.crossings[1, ]

##       Date Hour WestSide EastSide Day_of_week_name Day Month_name Weekend
## 1 1/1/2016    0       18        9              Fri   1        Jan       0

# Get the first five rows of bike.crossings
bike.crossings[1:5, ]

##       Date Hour WestSide EastSide Day_of_week_name Day Month_name Weekend
## 1 1/1/2016    0       18        9              Fri   1        Jan       0
## 2 1/1/2016    1       15        3              Fri   1        Jan       0
## 3 1/1/2016    2       11        6              Fri   1        Jan       0
## 4 1/1/2016    3        7        1              Fri   1        Jan       0
## 5 1/1/2016    4        2        0              Fri   1        Jan       0

# Get the third column of bike.crossings
# Note that we used the head command so that it doesn't print out everything
# But in general you wouldn't use it when accessing a column
head(bike.crossings[, 3])

## [1] 18 15 11  7  2  6
```

Now, let's take a look at how we can get all counts from January using the `which`. First, let's take a look at how `R` evaluates yes/no statements (also called booleans). Note that when testing for equality, we use two equals signs (==).

```
X <- c(1, 2, 1, 3, 4)
X == 1

## [1]  TRUE FALSE  TRUE FALSE FALSE

X > 2

## [1] FALSE FALSE FALSE  TRUE  TRUE
```

```
X <= 3

## [1]  TRUE  TRUE  TRUE  TRUE FALSE
```

Using the `which` command returns the index of the elements which evaluate to `TRUE`.

```
which(X == 1)

## [1] 1 3
```

So now, to get a quick summary of the January and May bike crossings, we can use the `summary` function.

```
summary(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")])

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.00   33.00   69.53   85.25  501.00

summary(bike.crossings$WestSide[which(bike.crossings$Month_name == "May")])

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.00   43.00   76.07   96.00  560.00
```

Alternatively, we could get each of the statistics individually. Note that `R` has a generic function for quantiles, so to get the first and third quartiles specifically, we specify what proportion of the data is is below the number we want. Thus, we include an additional argument of .25 for the first quartile, and .75 for the 3rd quartile. Also, in general, when we aren't just talking about 4th (quartiles), and are just talking about any proportion, we use the word quantile (hence the function name).

```
min(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")])

## [1] 0

quantile(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")], .25)

## 25%
##   9

median(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")])

## [1] 33

mean(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")])

## [1] 69.5336

quantile(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")], .75)

##   75%
## 85.25

max(bike.crossings$WestSide[which(bike.crossings$Month_name == "Jan")])

## [1] 501
```

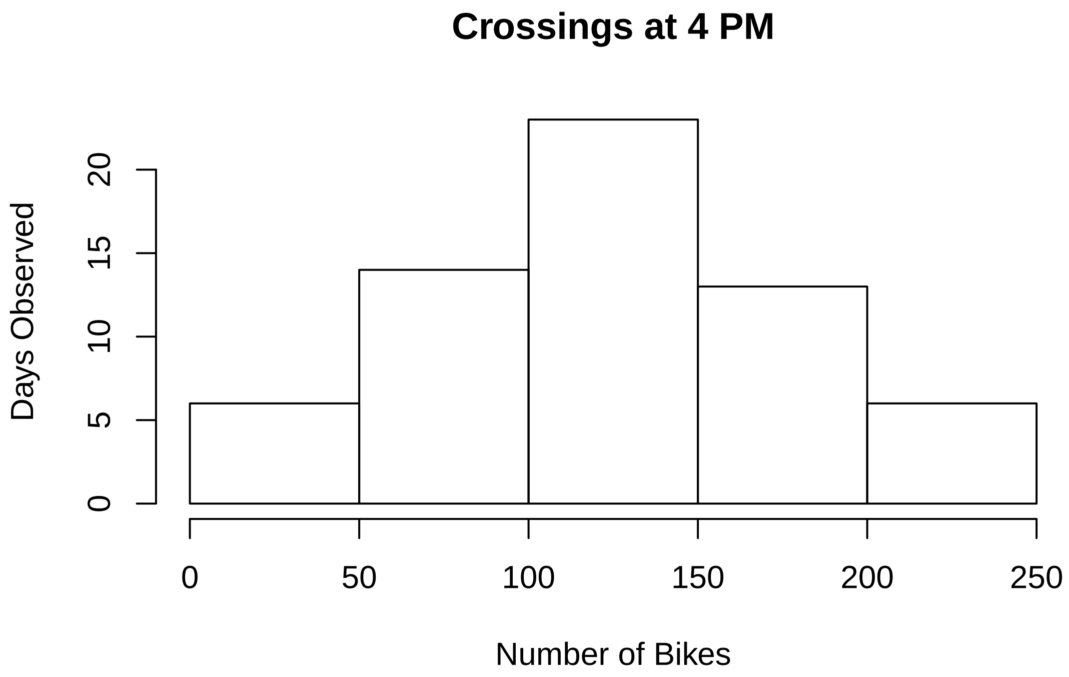
Questions

- Do we observe a difference between the months?
- Does this agree with what we might've expected? What might explain this difference?
- Is the data truly different or just different by chance? (No need to answer conclusively, just something to ponder)

Examining the Data in more detail

Let's take a look at the distribution of crossings at specific times of the day. First we examine all crossings at 4 PM. Note that since the hours are in 24 hour time, 4 PM corresponds to Hour 16.

```r
hist(bike.crossings$WestSide[which(bike.crossings$Hour == 16)],
     main = "Crossings at 4 PM", xlab = "Number of Bikes", ylab = "Days Observed")
```



Questions

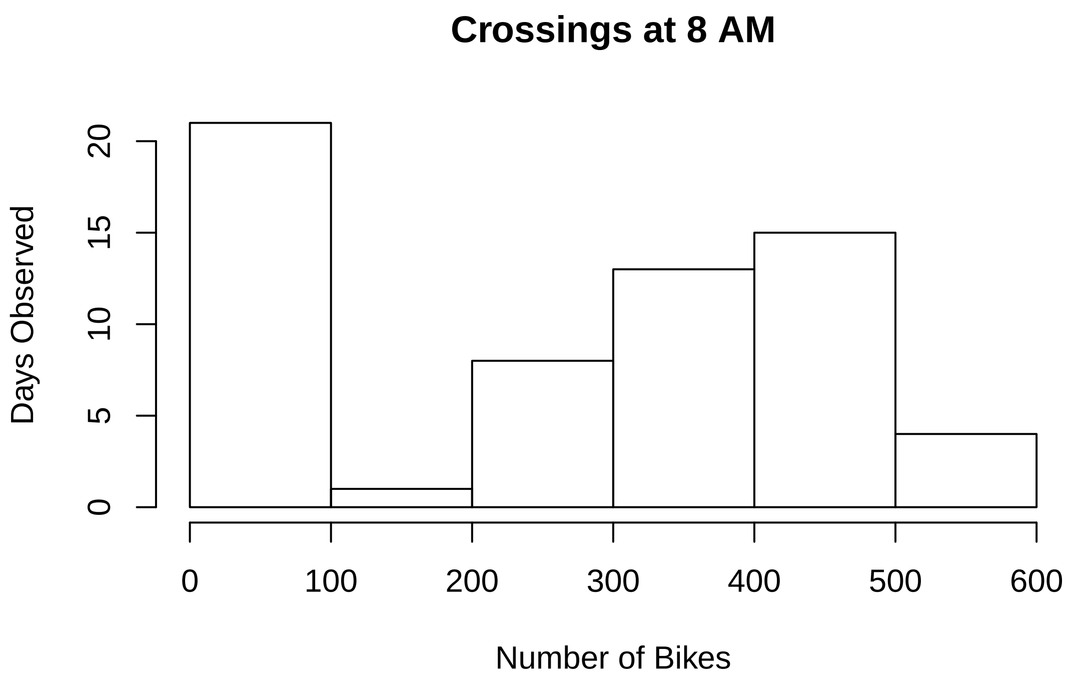
- Describe the shape of this distribution?
- How does this differ from the shape of all crossings for all hours above?
- Why might the shapes be drastically different?

Now let's take a look at the distribution of crossings at 8 AM.

```r
hist(bike.crossings$WestSide[which(bike.crossings$Hour == 8 )],
     main = "Crossings at 8 AM", xlab = "Number of Bikes", ylab = "Days Observed")
```

Questions

- How would you describe the distribution?
- How is this distribution different from the distribution of crossings at 4 PM?
- What might accout for the different shape? Recall what type of data we have



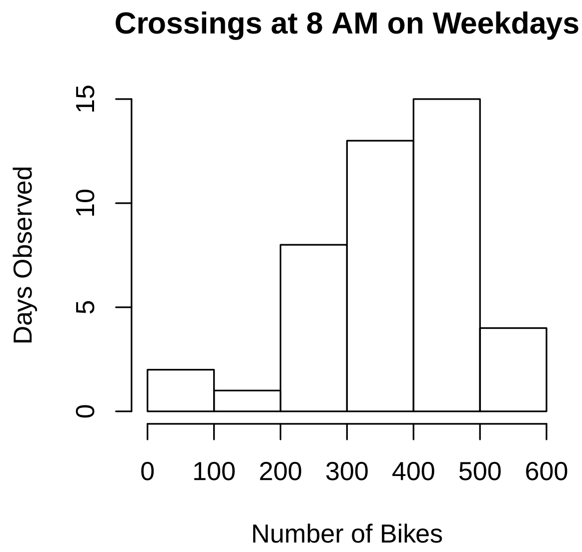
Let's see if further subsetting our data will change the distribution. We use the ampersand (&) to indicate ”and” statements. So if I want to test if A and B are true, I would use &. Note that this only returns true if both statements are true.

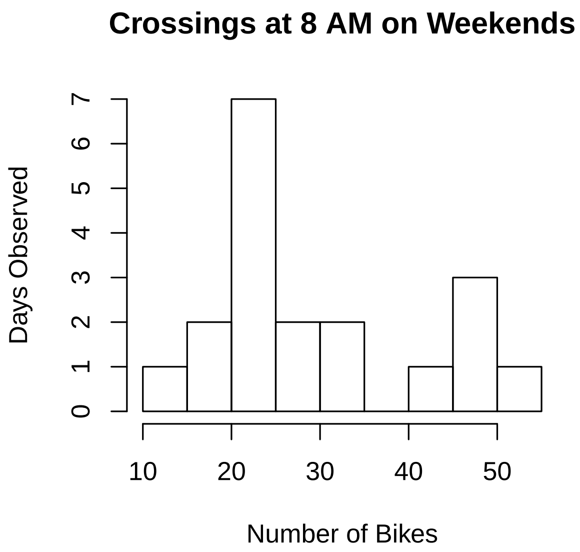
```
# Both statements are true
1 < 2

## [1] TRUE

2 < 3

## [1] TRUE

(1 < 2) & (2 < 3)

## [1] TRUE

# Only one statement is true
1 < 2

## [1] TRUE

3 < 2

## [1] FALSE

(1 < 2) & (3 < 2)

## [1] FALSE
```

So now let's take a look at the number of crossings at 8 AM, but also split out Weekends and Weekdays. The following command will get the indices of all rows which are at 8 AM **and** on a Weekday.

```
which(bike.crossings$Hour == 8 & bike.crossings$Weekend == 0 )
```

```
hist(bike.crossings$WestSide[which(bike.crossings$Hour == 8 & bike.crossings$Weekend == 0)],
     main = "Crossings at 8 AM on Weekdays",
     xlab = "Number of Bikes", ylab = "Days Observed")

hist(bike.crossings$WestSide[which(bike.crossings$Hour == 8 & bike.crossings$Weekend == 1)],
     main = "Crossings at 8 AM on Weekends",
     xlab = "Number of Bikes", ylab = "Days Observed")
```





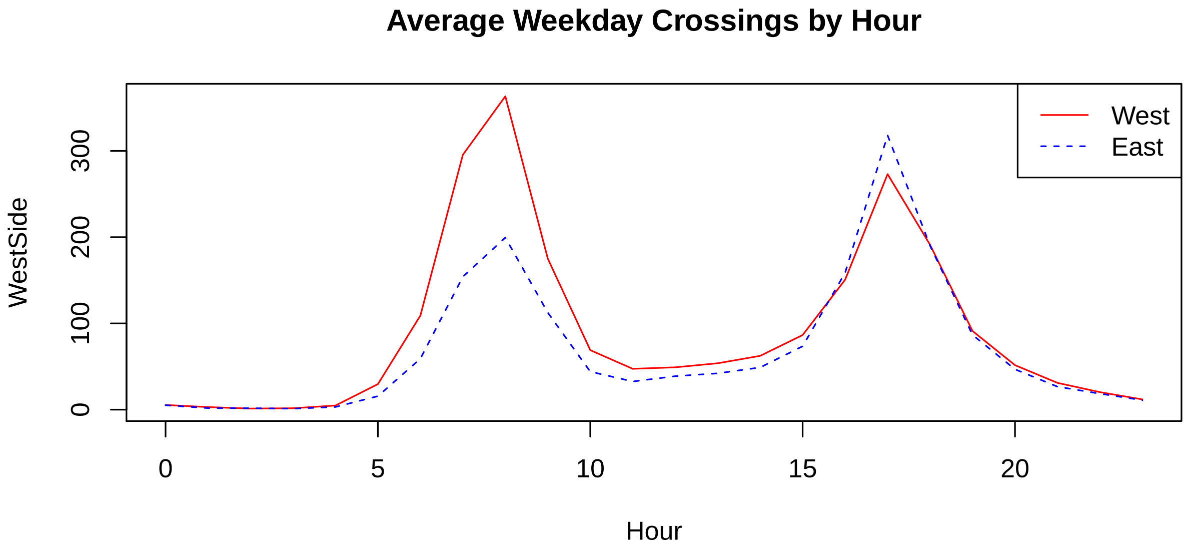
Questions

- How do weekend mornings differ from weekday mornings? Be careful to examine the scale on the x-axis.

East vs West

So far we have only examined data from the west side of the bridge. What might we be able to learn by looking at the east side of the bridge versus the West side of the bridge. We can plot the crossing by time for both the east side and the west side. For now, don't worry so much about this code (it's a bit complicated), but focus on the plot that is produced.

```
plot(aggregate(WestSide ~ Hour, data = bike.crossings,
               subset = which(Weekend == 0 ), FUN = mean),
     type = "l", col = "red", lty = 1, main =  "Average Weekday Crossings by Hour")

lines(aggregate(EastSide ~ Hour, data = bike.crossings,
                subset = which(Weekend == 0 ), FUN = mean),
      type = "l", col = "blue", lty = 2)

legend("topright", legend = c("West", "East"),
        lty = c(1,2), col = c("red", "blue"))
```



We know from before that there is We know from before that more bikes cross on the west side than the east side, but what else can we discover here?

Questions

- Assuming people are cycling to work and then returning home, does it appear that more people work in Fremont or Queen Anne?
- The peaks in the mornings seem higher than the peaks in the afternoon, does this make sense? How might we explain that observation?

Outliers

Let's take a look at all east side crossings in weekday afternoons (3 - 8) in May.

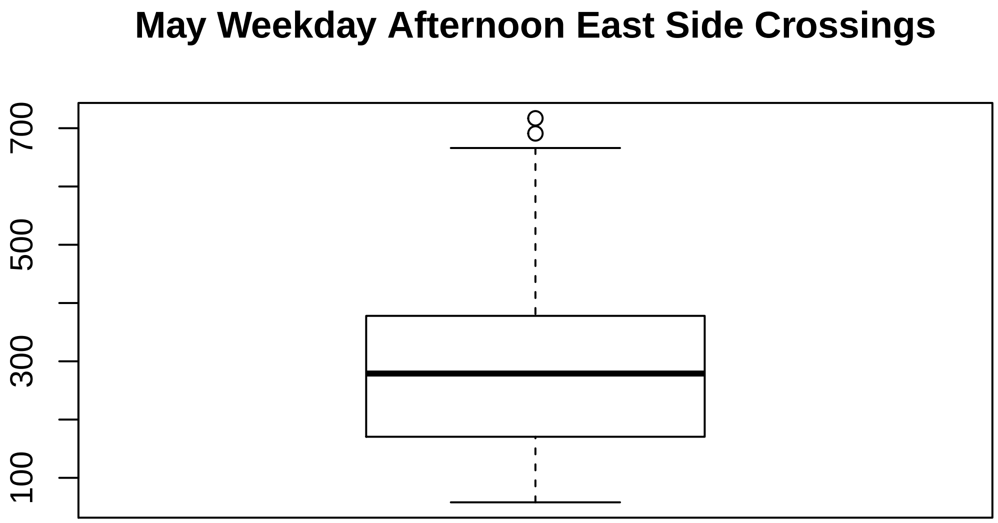
```
boxplot(bike.crossings$EastSide[which(bike.crossings$Hour < 20 &
                                        bike.crossings$Hour > 15 &
                                        bike.crossings$Month_name == "May" &
                                        bike.crossings$Weekend == 0)],
        main = "May Weekday Afternoon East Side Crossings")
```

As mentioned in class, in a box plot, outliers will be denoted by points on the graph beyond the "whiskers" of the boxplot. Although the definition of an outlier may vary by context, typically we define an outlier as

$$> Q3 + 1.5 \times IQR \tag{1}$$

or

$$< Q1 - 1.5 \times IQR \tag{2}$$



Where Q3 denotes the 3rd quartile, Q1 denotes the 1st quartile and IQR denotes the interquartile range (Q3 - Q1). It looks like we have 2 outliers in our plot above. Let's take a closer look.

We first get the indices of the two largest values

```
# order returns the ranks of each of the values
# we just want the indices of the two largest values
order(bike.crossings$EastSide, decreasing = T)[1:2]

## [1] 978 786
```

If we take a look at row 978, we can see that it corresponds to 5:00 PM on May 10, which was Global Bike to Work Day. The other outlier is May 2, the first day of Bike to Work Month. It looks like these two intiatives worked!

```
bike.crossings[978, ]

##          Date Hour WestSide EastSide Day_of_week_name Day Month_name
## 978 5/10/2016   17      240      717              Tue  10        May
##     Weekend
## 978       0

bike.crossings[786, ]

##         Date Hour WestSide EastSide Day_of_week_name Day Month_name
## 786 5/2/2016   17      261      691              Mon   2        May
##     Weekend
## 786       0
```

Let's take a look at how the mean, median, standard deviation and IQR are affected by removing these two outliers. Note we use the `sd` function to calculate standard deviation and `IQR` function to calculate the IQR.

```
## With the outlier

east.side.afternoon <- bike.crossings$EastSide[which(bike.crossings$Hour < 20 &
                                        bike.crossings$Hour > 15 &
                                        bike.crossings$Month_name == "May" &
                                        bike.crossings$Weekend == 0)]

order(east.side.afternoon, decreasing = T)[1:2]

## [1] 26  2

mean(east.side.afternoon)

## [1] 306.6932

mean(east.side.afternoon[-c(26,2)])

## [1] 297.4535

median(east.side.afternoon)

## [1] 279

median(east.side.afternoon[-c(26,2)])

## [1] 275

sd(east.side.afternoon)

## [1] 164.233

sd(east.side.afternoon[-c(26,2)])

## [1] 154.2808

IQR(east.side.afternoon)

## [1] 204.75

IQR(east.side.afternoon[-c(26,2)])

## [1] 201.25
```

0.1 Question

- Does the Mean or median change more when the outliers are removed?
- Does the SD or IQR change more when the outliers are removed?

Lab Assignment

Please see the lab assignment posted on catalyst. The file should be submitted to the catalyst dropbox by Monday June 27 11:59 PM.