Katherine Olson

STA 141 Assignment One Part II

Question 1

Find at least 3 *types* of anomalies in the data. Provide succinct justification for identifying them as anomalies. Then correct the corresponding observations appropriately, again providing justification. What impact does this have on analyzing the data?

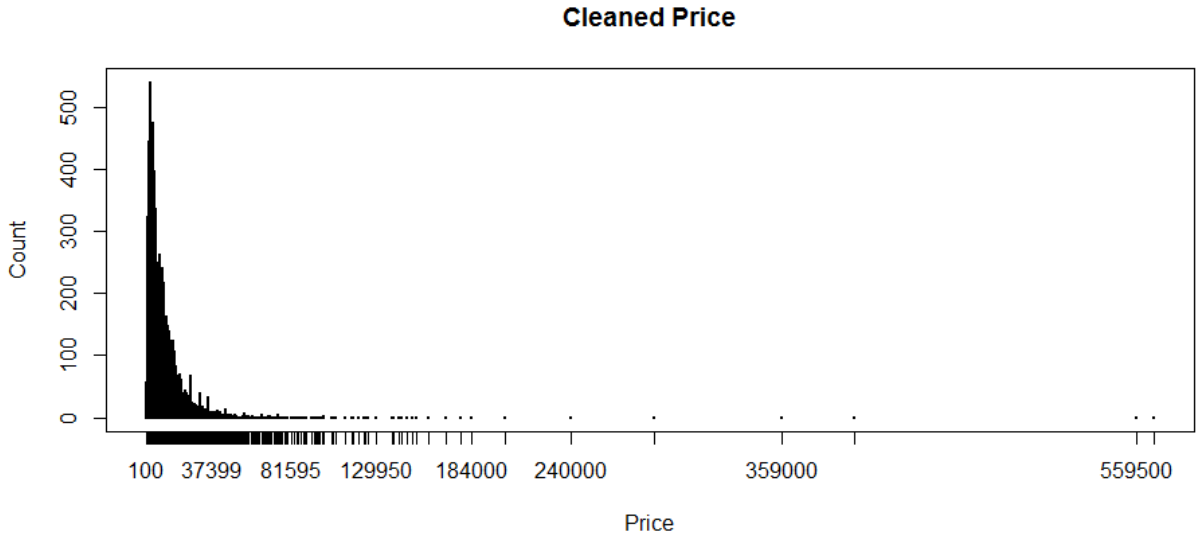
Anomaly 1: High and low outliers in price

I remembered from part I that price had a lot of high and low outliers. To recheck this, I graphed the original values as seen below.

This plot looks very wrong. Thinking this is probably due to outliers, I looked at the mean and median. The mean price is $49,491 and the median price is $6,700. There is a huge difference between the two. The graph above shows a range of prices from $1 to $6,000,300,000. Neither the maximum or minimum are reasonable prices for a car. This must be an anomaly.

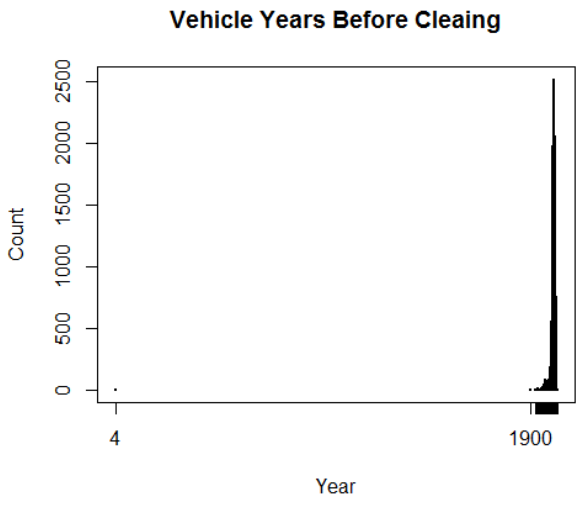
I decided to start with the $1 price. There are 612 vehicles with this price. Something is definitely wrong here. I used gsub() to look though each body with a price of $1 for a dollar sign followed by a large number. I found a variety of patterns and kept looking through the bodies until I could spot no more patterns. I was left with 145 vehicles. I changed these to NA. I repeated this process for cars with a price below $100 and then for cars above $900,000. There were only three vehicles above $900,000. One was a typo that I fixed and the other two gave no clues to the real price so they were set to NA. I fixed most of the outliers and reached a point

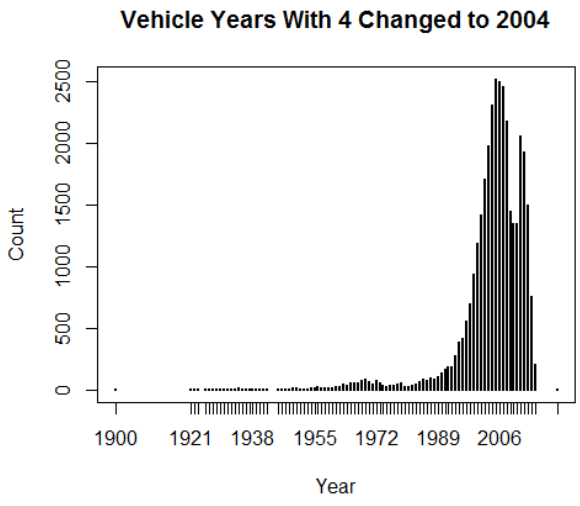
where cleaning the data anymore would be more effort than it would be worth, so I stopped. The final plot is below. The distribution is skewed right with most cars below $40,000.



Anomaly 2: Year

I decided to use year as my second anomaly because there are a lot of strange values. There are a few years like 4 and 2022 that do not make sense and there are a lot of cars with 2016 as the year, but a different year in the title. In part I, I tackled this the long way, by going through each title and making the changes one by one, but in class today, Duncan showed us a much faster way using gsub() that I will be using. I started by looking at the years on a plot to get a feel for where the problems are.



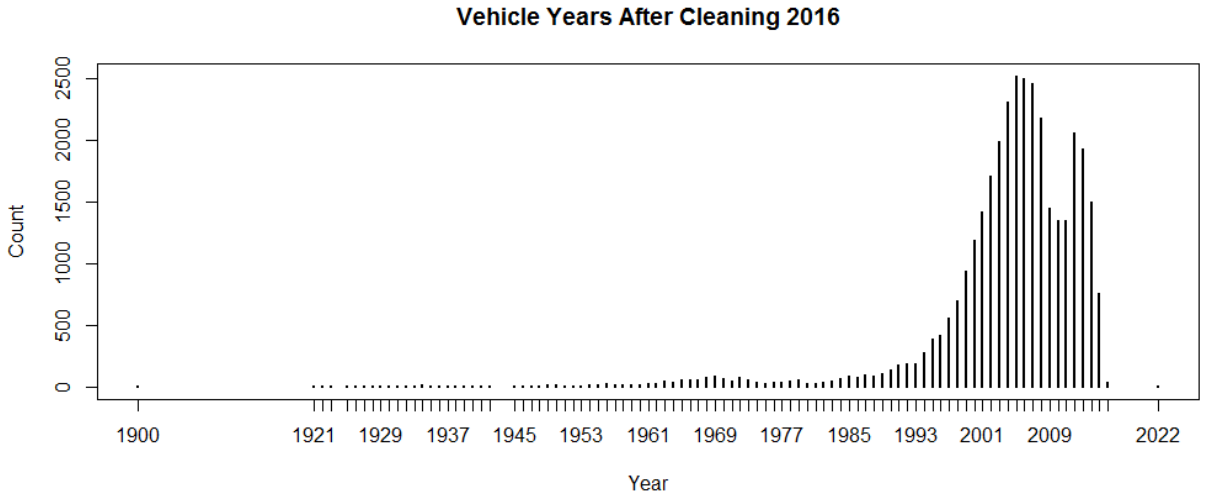


Looking at the plots on the previous page, the plot on the left is a plot of year before any changes were made. This plot shows a clear outlier of 4, upon further investigation this was a typo and I changed the year to 2004. The plot on the right shows the new distribution of years and changing the 4 made the plot a lot clearer.

I decided to start the cleaning process by working on the vehicles with a year of 2016. There are 206 and of these cars. It does not make sense that there would be so many new cars for sale on the website. Looking through some of the titles, it is clear that a lot of these 2016s are a mistake and the correct year is listed in the title. Using gsub(), I searched through all of the titles for anything in the 1900s and 2000s to 2010s. After making these changes, there are 78 vehicles left with a year of 2016. Looking through the titles again, there were a few that I changed individually with a year of 98 or 03 that gsub() did not catch. I did the same for header. This left 67 vehicles.

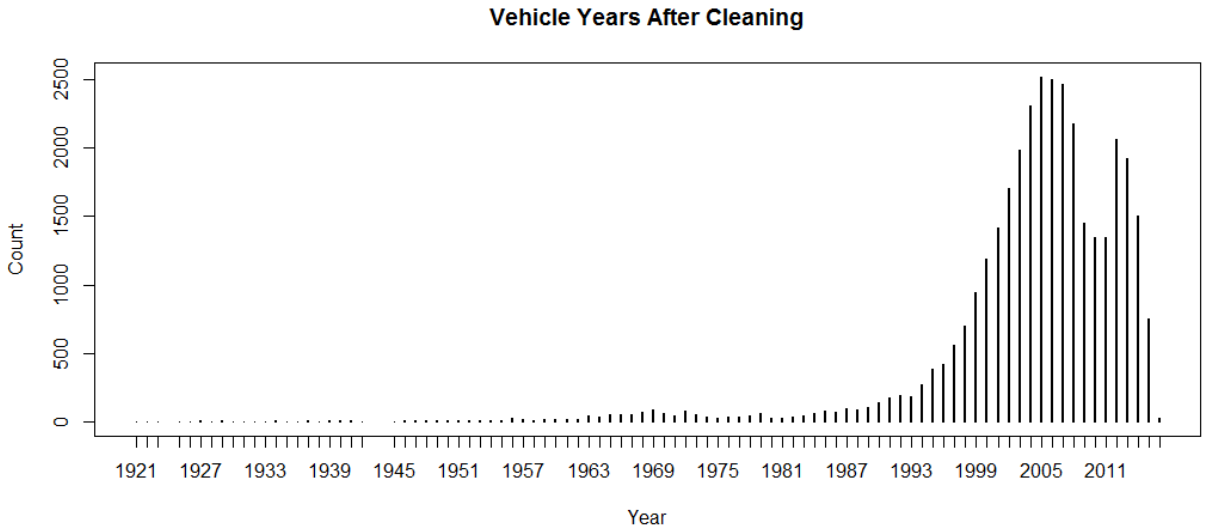
While looking through the titles, there were a few that caught my eye as repeats or strange. The strange ones were titles that sounded like they weren't selling cars. I went through and deleted or changed these using some of my code from part I. This resulted in 43 vehicles with a year of 2016 left. I then decided to look through some of the bodies of the cars left to see if another gsub() though body would make any difference. There were many bodies with years other than 2016, so I used gsub() again and was left with 31 years of 2016. I decided that it was time to stop data cleaning. Going from 206 to 31 was good enough and the rest of the titles seem like they are actually 2016 cars. Looking into each of these would be more effort than the payoff would be worth.

My next step was to plot the data again to make sure gsub() did not pick up anything like 20000 instead of 2000. The graph is below. There appear to be no major mistakes.



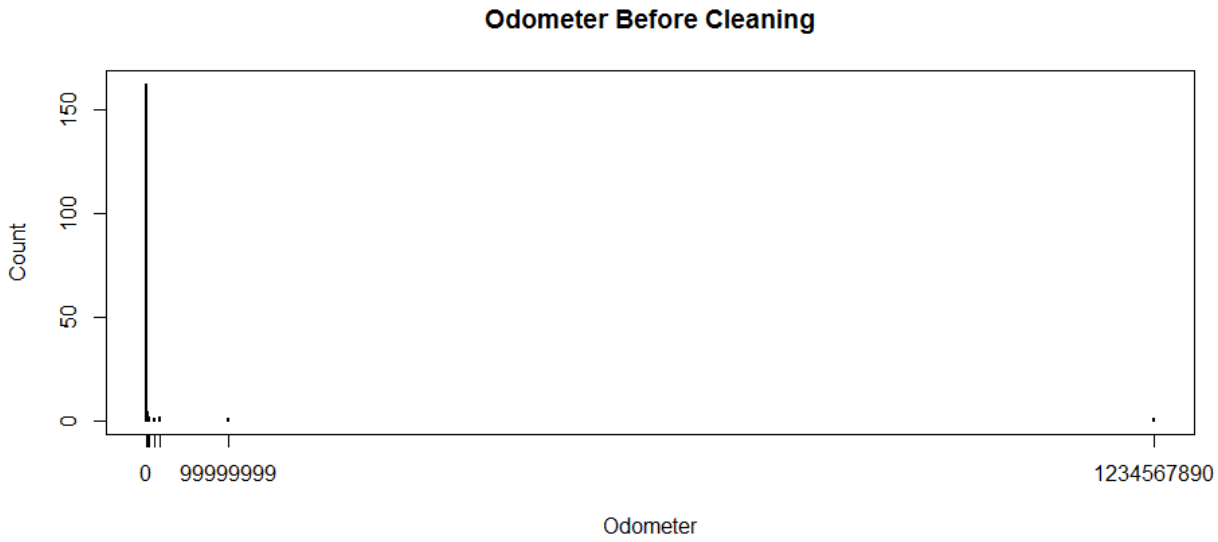
Looking at the graph above, there appear to be a few high and low outliers. I investigated those rows, and made changes. I am now pleased with the level of which the data has been cleaned.

There are most likely many more mistakes throughout the data, but the effort I would put into looking though every year would not be worth the result. So it is time to stop. The final plot of the cleaned data is below. The distribution looks much better than the original plot. The data is skewed left with most years between 1995 and 2015.



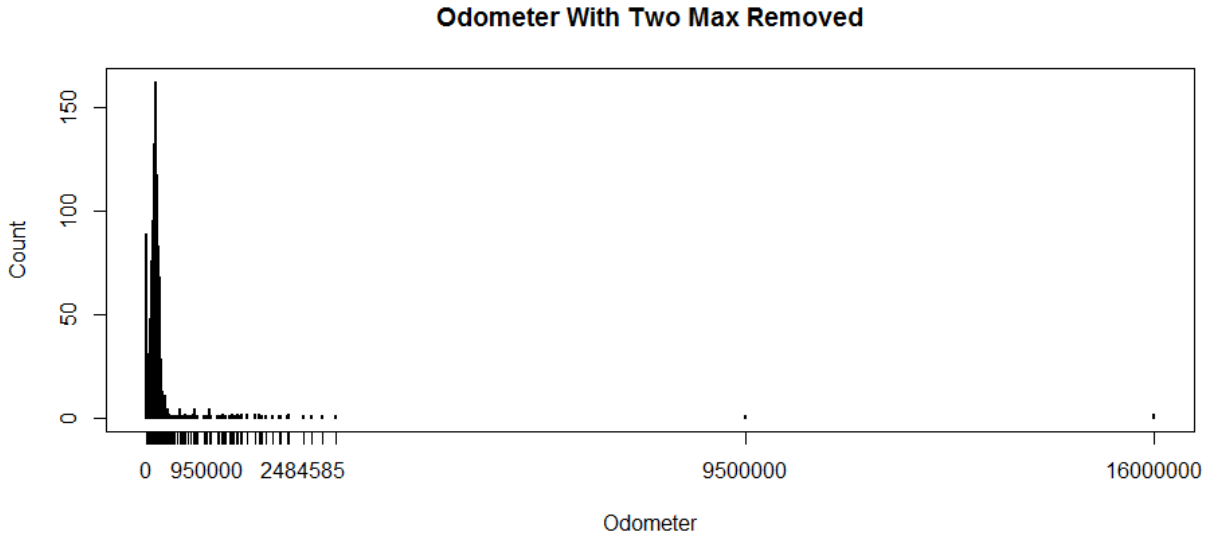
Anomaly 3: Odometer

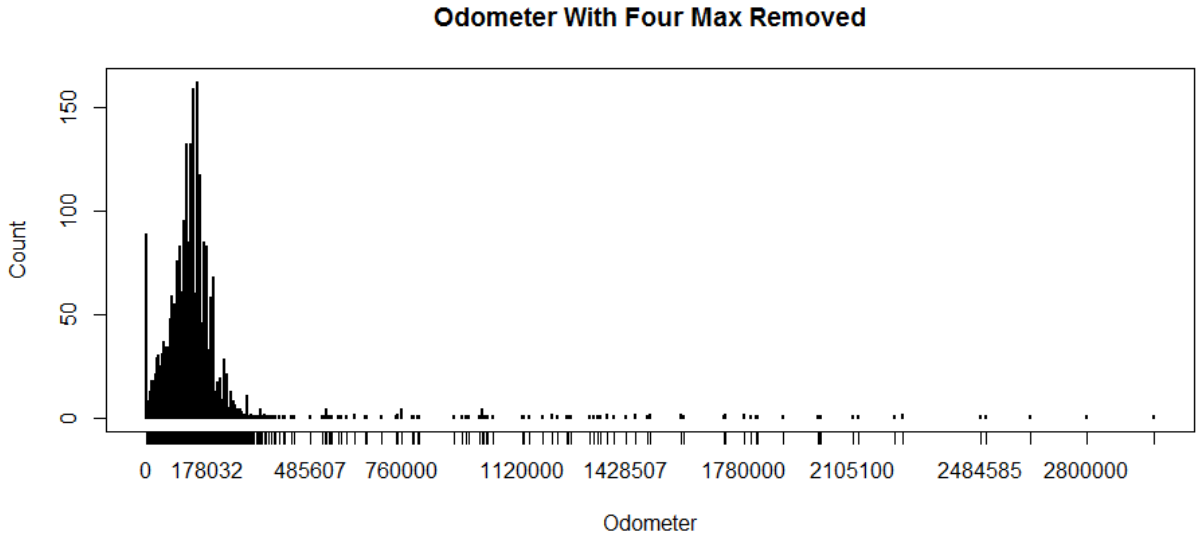
Looking at a plot of the unchanged odometer values below, there are a few giant outliers throwing the distribution off. There also appear to be a lot of cars with an odometer of zero, this makes sense as long as those values are new cars, which might not be the case.



I started the cleaning process by looking at the rows of the two largest values on the graph above. There were no hints in the rows to what the correct odometer might be, so they were changed to

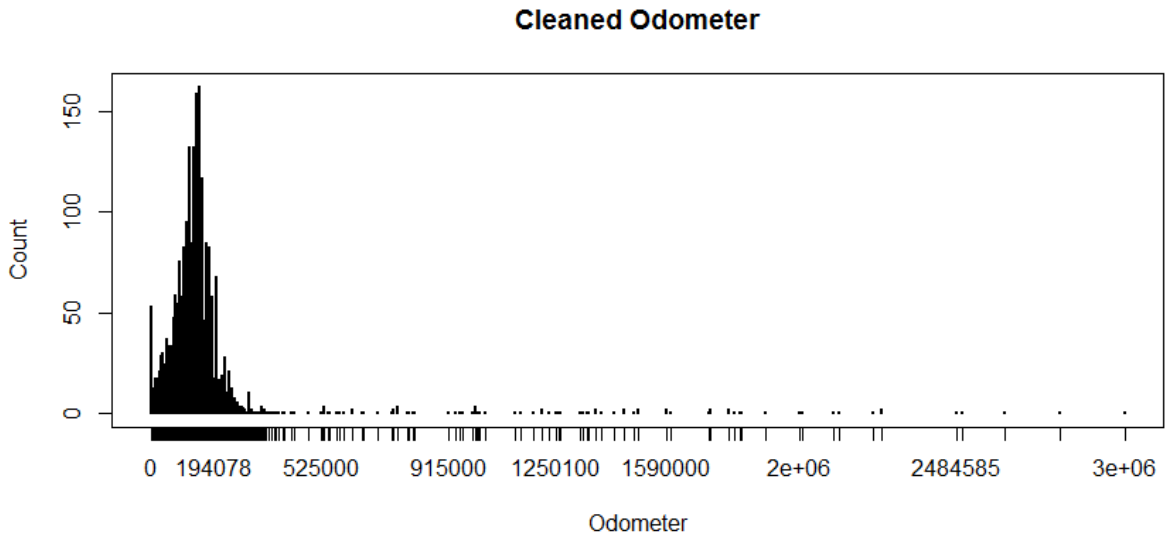
NA. I plotted the data again and the result is below. This plot shows two more clear high outliers. I repeated the same process and changed them to NA. The resulting plot is also below.





Now that there are no max values way off on their own making the distribution hard to see, I want to check the condition and year of the vehicles with an odometer of zero. The condition should be new and the year should be 2015 or 2016, but I do not know how clean this area of data is. There are 89 vehicles with an odometer of zero. Most of the conditions look right, but there are a few used. The years are mostly lower than 2015. I decided to leave the odometer of

2015 cars at zero and investigate the bodies of the rest. Only one body had information about the odometer and I changed the rest to NA. I repeated the same process for vehicles with an odometer of one. At this point, I decided that it was time to stop cleaning. It would take me more time than it would be work to go through the rest of the low and high values. The plot with the cleaned data is below. The plot is skewed right.



Changing the data like this for all three anomalies makes the true distribution more clear. We risk losing outliers that may be important or have meaning, but those outliers make plots much more difficult to see and mess with the mean.

Question 2

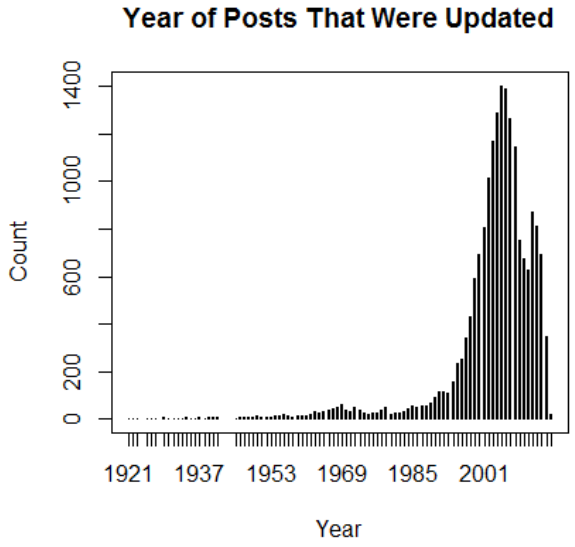
Find at least 3 interesting insights/characteristics/features illustrated by the data. Explain in what way these insights are interesting (to whom? why?) and provide evidence for any inference/conclusions you draw. How generalizable are these insights to other vehicle sales data?

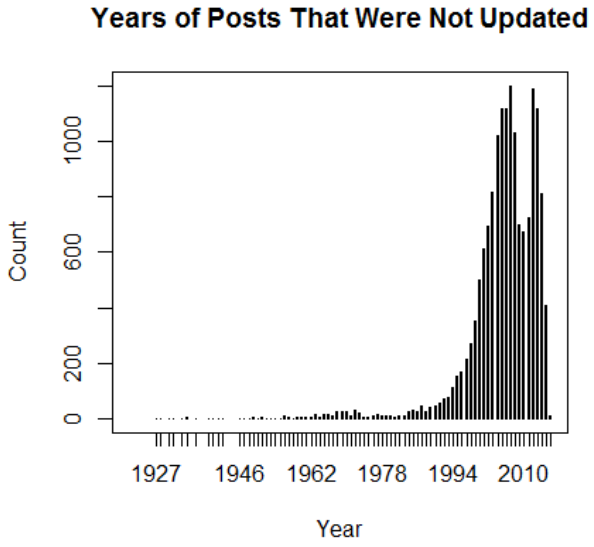
I approached this question by working with the three variables I used as anomalies and made changes to in question one. I tried plotting different variables and seeing what I could find or interest.

Insight 1:

I subsetted the data into one group where the post had been updated and another group where the post had not been updated. A post is most often updated to lower the price. A post that was updated indicates that the vehicle is not selling. I graphed the years of these subsets on two plots and expected to find a difference. I was thinking that newer cars would sell quickly, but looking at the plots on the next page, that is not the case. The two graphs are barely any different. This supports a conclusion that for the website the data was taken from, the year of a car does not

affect how quickly it sells. I think this can be interesting to anyone looking to sell a car online. They should not be discouraged if they think no one will buy their old car. This data was taken from a specific website, so it cannot be used to predict the relationship between year and if a post was updated or not for the whole population. It can only be used to guess, knowing that the guess could be completely wrong because of different populations. The data could have bias. There are only seven cities and there is most likely some bias in the type of person that would sell their car on this website that separates them from the whole population. The variables posted and updated are not very clear on why the post was updated. This could indicate trouble selling a vehicle, but we do not know this for sure. So anyone trying to generalize this information should be careful and aware that making inferences from this data set could lead to wrong inferences.



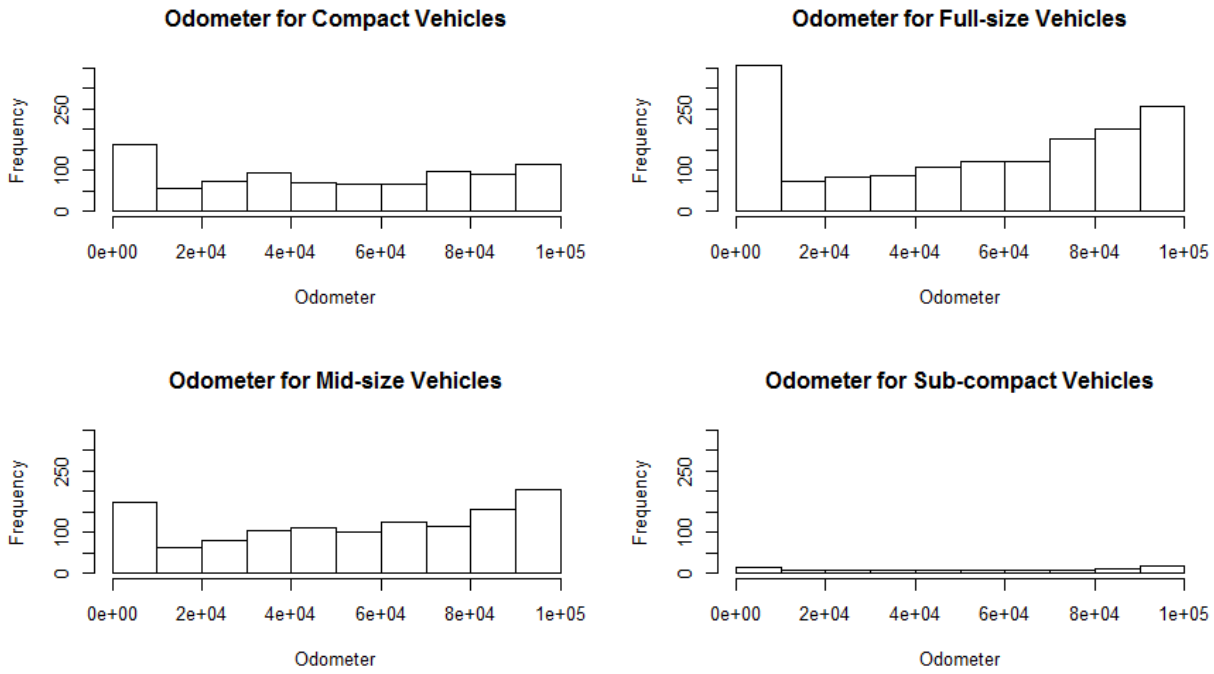


Insight 2:

For this insight, I looked at the relationship between odometer and vehicle size. I was curious to find out what size cars have gone more miles and if there is a difference between the sizes. There are four categories of cars size, so I plotted four histograms all on the same scale showing the distribution of odometer. The plots are on the next page. The distributions for mid-size, sub-compact, and compact are all fairly flat, while the distribution for full-size is mostly skewed left, but the furthest bar to the left is way taller than the rest. This left bar brings the distribution closer to a symmetric u-shape. So for full-size vehicles, there are a lot of high and low values, but not a lot in the middle.

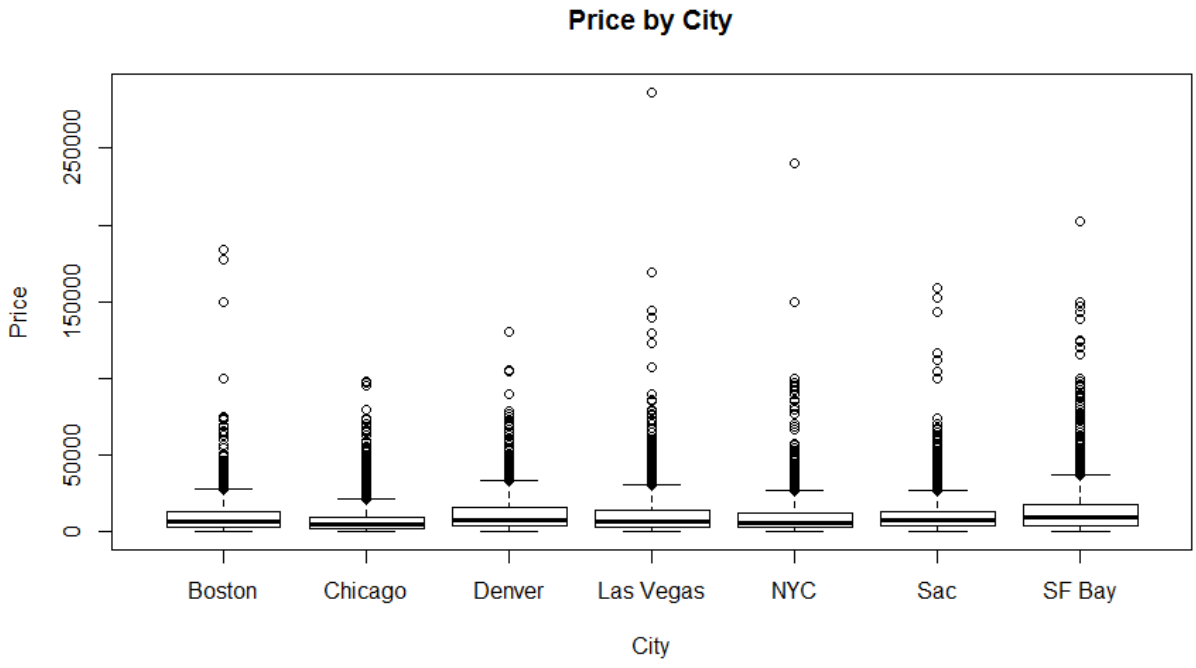
Overall, there is not a huge difference in odometer for each size of vehicle. It is useful to know that the previous vehicle owners of the vehicles in this data set did not prefer a certain size car for driving more or less miles. So for anyone deciding what car size to get and trying to factor in

the amount of miles they expect to drive, they can learn that if they are buying a car on this website, they do not need to worry about how many miles they will drive. Of course there are many other factors in the size of a car a person buys. They should just beware that this can only be said for this set of data, and might not apply to others.



Insight 3:

For this insight, I looked at the relationship between city and price. I was expecting to find a difference of some sort. I was expecting some cities to have more expensive cars than others.



Looking at the plot on the previous page, the price for each city is not that different. It is even hard to tell which cities have the higher or lower prices. This can be useful information to anyone buying a car from this website. They do not need to worry about the closest city to them being the most expensive. There is no need for them to travel to another city to buy a car. The difference in price between cities is not different enough. This makes finding a car easier since they can just look at cars in their city instead of searching though all the cities thinking their effort might produce a cheaper car in a farther away city. Of course, this is true overall and they could still find a cheaper car in another city. They just need to consider that all of that effort is most likely for nothing. This only applies to this data set, it could be true for the population, but we cannot say that. If someone wants to see this plot and decide that price varies very little for all cars in all cities, that would be very dangerous. They could be right, but they could also be very wrong.

Code Appendix

Question 1

```
# # # # # Anomaly # 1: Price
# What are we working with:
plot(table(vposts$price), main = "Price Before Cleaning", xlab = "Price", ylab = "Count")
# Range" $1 to $6,000,300,000 (neither are reasonable car price)
mean(vposts$price, na.rm = TRUE)
median(vposts$price, na.rm = TRUE)
table(vposts$price == 1) # 612
# Looking at body for proper price:
vposts[ !is.na(vposts$price) & vposts$price == 1, c("body")]
# Using gsub() to search for prices with 4 digits
locs = which(vposts$price == 1)
w = grepl(".*\\$[0-9][0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9][0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price == 1) #  499 left
# Using gsub() to search for prices with a comma in the thousands place:
locs = which(vposts$price == 1)
w = grepl(".*\\$[0-9],[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9],[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price == 1) # 482 left
# Using gsub() to search for prices with a comma in the  ten thousands place:
locs = which(vposts$price == 1)
```

```
w = grepl(".*\\$[0-9][0-9],[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9][0-9],[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price == 1) # 475 left
# Looking for more patterns:
vposts[ !is.na(vposts$price) & vposts$price == 1, c("body")]
locs = which(vposts$price == 1)
# Show bodies with "asking" in them:
w = which(grepl(".*asking.*", vposts$body[locs]))
print(vposts[w,]$body)
vposts[6,]$price = 28996
vposts[66,]$price = 13991
vposts[471,]$price = 184000
table(vposts$price == 1) # 475 left
# Can't find any more patterns, changing the rest to NA
vposts[!is.na(vposts$price) & vposts$price == 1, ]$price = NA
# Now all $1 are dealt with, finding other min values:
loc = which.min(vposts$price)
print(vposts[loc, ]$body) # Tells us nothing about price, set to NA
vposts[loc, ]$price = NA
# Looking for patterns of vehicles with price < 100
vposts[ !is.na(vposts$price) & vposts$price < 100, c("body")] # 234
# Same as before:
locs = which(vposts$price < 100)
w = grepl(".*\\$[0-9],[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9],[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 222 left
locs = which(vposts$price < 100)
w = grepl(".*\\$[0-9][0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9][0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 169 left
locs = which(vposts$price < 100)
w = grepl(".*\\$[0-9][0-9],[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9][0-9],[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 158 left
# Looking one more time for patterns:
vposts[ !is.na(vposts$price) & vposts$price < 100, c("body")] # 234
```

```
# Those patterns:
locs = which(vposts$price < 100)
w = grepl(".*\\$[0-9].[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9].[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 151 left
locs = which(vposts$price < 100)
w = grepl(".*\\$[0-9][0-9].[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$[0-9][0-9].[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 149 left
locs = which(vposts$price < 100)
w = grepl(".*\\$ [0-9][0-9].[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$ [0-9][0-9].[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 147 left
locs = which(vposts$price < 100)
w = grepl(".*\\$ [0-9],[0-9][0-9][0-9].*", vposts$body[locs])
vposts$price[locs[w]] = as.integer(gsub(".*\\$ [0-9],[0-9][0-9][0-9].*", "\\1",
        vposts$body[locs[w]]))
table(vposts$price < 100) # 145 left
vposts[ !is.na(vposts$price) & vposts$price < 100, c("body", "header")]
# Can't find any more patterns, setting rest to NA:
vposts[!is.na(vposts$price) & vposts$price < 100, ]$price = NA
# Repeating process for cars above 900000
vposts[ !is.na(vposts$price) & vposts$price > 900000, c("body")] # 234
# Code from Q8 part I:
maxPrice = print(max(vposts$price, na.rm = TRUE)) # 600030000
maxPriceRow = print(subset(vposts, vposts$price == maxPrice))
newPrice = (6000 + 30000) / 2
maxLocation = which.max(vposts$price)
vposts[maxLocation,]$price = newPrice # Fixed
vposts[ !is.na(vposts$price) & vposts$price > 900000, c("body", "description", "price")]
# 2 above 900,000 and no clues about real price, changing to NA
vposts[!is.na(vposts$price) & vposts$price > 900000, ]$price = NA
max(vposts$price, na.rm = TRUE)
plot(table(vposts$price), main = "Cleaned Price", xlab = "Price", ylab = "Count")

# # # # # Anomaly # 2: Year
par(mfrow = c(1,2))
```

```
# Seeing what we are working with:
plot(table(vposts$year), main = "Vehicle Years Before Cleaing", xlab = "Year", ylab = "Count")
# Right away it is clear that 4 is wrong and removing that will make the plot easier to see
print(vposts[vposts$year == 4,])
vposts[vposts$title == "argolic eni-04 JEeP wraNgler Clean lEATHeR - $2532 (chicago)",]$year
    = 2004
plot(table(vposts$year), main = "Vehicle Years With 4 Changed to 2004", xlim = c(1900, 2022),
    xlab = "Year", ylab = "Count")
table(vposts$year == 2016) # 206 cars with 2016
vposts[ !is.na(vposts$year) & 2015 - vposts$year == -1, c("year", "title")] # For many, correct
    # year is in title
# Using gsub to search through titles: (Got this code from lecture 10/8)
locs = which(vposts$year == 2016)
w = grep(".*19[0-9][0-9].*", vposts$title[locs])
vposts$year[locs[w]] = as.integer(gsub(".*19[0-9][0-9].*", "\\1", vposts$title[locs[w]]))
table(vposts$year == 2016) # 164 left
locs = which(vposts$year == 2016)
w = grep(".*20[0-1][0-9].*", vposts$title[locs])
vposts$year[locs[w]] = as.integer(gsub(".*20[0-1][0-9].*", "\\1", vposts$title[locs[w]]))
table(vposts$year == 2016) # 78 left
# Looking at titles again and changing a few with no pattern:
vposts[ !is.na(vposts$year) & 2015 - vposts$year == -1, c("year", "title")]
vposts[vposts$title == "01 LINCOLN TOWN CAR reduced!!!!! - $2500 (Walpole)",]$year =
    2001
vposts[vposts$title == "03 maxima 6 speed $2100 obo - $2100 (carpentersville)",]$year = 2003
vposts[vposts$title == "02 Expedition - $4500 (aurora)",]$year = 2002
vposts[vposts$title == "98 Chrysler Town and Country 7 passenger leather seats only 98k miles -
    $1800 (Denver)",]$year = 1998
vposts[vposts$title == "96 mustang gt,5 speed,4.6 - $2300 (Sacramento,antelope)",]$year = 1996
vposts[vposts$title == "98 Honda civic DX - $3000 (Elk Grove)",]$year = 1998
vposts[vposts$title == "03 Lexus es300 $3900 obo - $3900 (Sacramento)",]$year = 2003
vposts[vposts$title == "95 PLYMOUTH VOYAGER - $1300 (fairfield / vacaville)",]$year =
    1995
table(vposts$year == 2016) # 70 left
# Same for header:
vposts[ !is.na(vposts$year) & 2015 - vposts$year == -1, c("year", "header")]
vposts[vposts$header == "2016 98 frontier",]$year = 1998
vposts[vposts$header == "2016 1991 ford bronco ii",]$year = 1991
vposts[vposts$header == "2016 Cobra mystic 1996",]$year = 1996
table(vposts$year == 2016) # 67 left
```

```
# Strange sounding titles:
print(vposts[vposts$header == "2016 Variety",]) # Selling many cars on one post: delete
vposts = vposts[- (which(vposts$header == "2016 Variety")), ]
print(vposts[vposts$header == "2016 New",]) # Window cleaning: delete
vposts = vposts[- (which(vposts$header == "2016 New")), ]
print(vposts[vposts$header == "2016 2015",]) # Selling rims: delete (four)
vposts = vposts[- (which(vposts$header == "2016 2015")), ]
print(vposts[vposts$header == "2016 All years makes and models",]) # Want you to sell them
        # your car: delete (3)
vposts = vposts[- (which(vposts$header == "2016 All years makes and models")), ]
print(vposts[vposts$header == "2016 ALL CARS TRUCKS SUVS",]) # Ad for place that sells
        # cars: delete (2)
vposts = vposts[- (which(vposts$header == "2016 ALL CARS TRUCKS SUVS")), ]
print(vposts[vposts$header == "2016 LOW DOWN PAYMENTS",]) # Ad: delete
vposts = vposts[- (which(vposts$header == "2016 LOW DOWN PAYMENTS")), ]
print(vposts[vposts$header == "2016 Caddilac",]) # Selling 44 cars: delete
vposts = vposts[- (which(vposts$header == "2016 Caddilac")), ]
print(vposts[vposts$header == "2016 Dodge, ford, Chevy",]) # Not a car: delete
vposts = vposts[- (which(vposts$header == "2016 Dodge, ford, Chevy")), ]
print(vposts[vposts$header == "2016 any",]) # delete
vposts = vposts[- (which(vposts$header == "2016 any")), ]
print(vposts[vposts$header == "2016 All",]) # delete
vposts = vposts[- (which(vposts$header == "2016 All")), ]
print(vposts[vposts$header == "2016 rr",]) # delete
vposts = vposts[- (which(vposts$header == "2016 rr")), ]
print(vposts[vposts$header == "2016 Acura ILX w/Technology Plus Pkg",]) # delete
vposts = vposts[- (which(vposts$header == "2016 Acura ILX w/Technology Plus Pkg")), ]
print(vposts[vposts$header == "2016 Acura ILX w/Premium A-SPEC Package",]) # delete
vposts = vposts[- (which(vposts$header == "2016 Acura ILX w/Premium A-SPEC Package")), ]
print(vposts[vposts$title == "1972 VW Super Beetle - $4000",]) # delte 2, change one's year to
        # 1972
locs = which(vposts$title == "1972 VW Super Beetle - $4000")
vposts = vposts[- locs[1:2], ]
vposts[vposts$title == "1972 VW Super Beetle - $4000",]$year = 1972
table(vposts$year == 2016) # 43 left
# Looking through body to decide if another gsub() would make a difference
vposts[ !is.na(vposts$year) & 2015 - vposts$year == -1, c("body")] # Yes
locs = which(vposts$year == 2016)
w = grep(".*19[0-9][0-9].*", vposts$body[locs])
vposts$year[locs[w]] = as.integer(gsub(".*19[0-9][0-9].*", "\\1", vposts$body[locs[w]]))
```

```
table(vposts$year == 2016) # 39 left
locs = which(vposts$year == 2016)
w = grep(".*200[0-9].*", vposts$body[locs])
vposts$year[locs[w]] = as.integer(gsub(".*200[0-9].*", "\\1", vposts$body[locs[w]]))
table(vposts$year == 2016) # 34 left
locs = which(vposts$year == 2016)
w = grep(".*201[0-5].*", vposts$body[locs])
vposts$year[locs[w]] = as.integer(gsub(".*201[0-5].*", "\\1", vposts$body[locs[w]]))
table(vposts$year == 2016) # 31 left
# From 206 to 31, time to stop. Look at plot to check if gsub() made any big mistakes:
par(mfrow = c(1,1))
plot(table(vposts$year), main = "Vehicle Years After Cleaning 2016", xlab = "Year", ylab =
        "Count")
# Looking at cars older than 100 years:
vposts[ !is.na(vposts$year) & 2015 - vposts$year > 99, c("header", "description", "title")]
print(vposts[vposts$header == "1900 CAR",]) # delete - many posts of ad to buy cars
vposts = vposts[- (which(vposts$header == "1900 CAR")), ]
print(vposts[vposts$header == "1900 Wheels",]) # Wheels: Delete
vposts = vposts[- (which(vposts$header == "1900 Wheels")), ]
# Looking at cars newer than 2015:
vposts[ !is.na(vposts$year) & 2015 - vposts$year < -1, c("title")] # Piazza
# One car: 2022 Honda Odyssey
print(vposts[vposts$title == "Check Out This Spotless 2022 Honda Odyssey with 117,102 Miles
        - $6999 (Jamaica)",])
# Found website, compared picture to different years of that car
vposts[vposts$title == "Check Out This Spotless 2022 Honda Odyssey with 117,102 Miles -
        $6999 (Jamaica)",]$year = 2015
# Time to stop cleaning, final plot:
plot(table(vposts$year), main = "Vehicle Years After Cleaning", xlab = "Year", ylab = "Count")


# # # # # Anomaly # 3:

plot(table(vposts$odometer), main = "Odometer Before Cleaning", xlab = "Odometer", ylab =
        "Count")
# 99999999 and 1234567890 are two clear large outliers
print(subset(vposts, vposts$odometer == 1234567890))
# No hints at the real odometer: make NA
vposts[which(vposts$odometer == 1234567890 & !is.na(vposts$odometer)), ]$odometer = NA
print(subset(vposts, vposts$odometer == 99999999))
# No hints at the real odometer: make NA
```

```
vposts[which(vposts$odometer == 99999999 & !is.na(vposts$odometer)), ]$odometer = NA
plot(table(vposts$odometer), main = "Odometer With Two Max Removed", xlab = "Odometer",
        ylab = "Count")
# 2 more high values: 9500000 and 16000000
print(subset(vposts, vposts$odometer == 16000000))
# 2 entries - lowered price by $100, no hints on actual odometer: NA
vposts[which(vposts$odometer == 16000000 & !is.na(vposts$odometer)), ]$odometer = NA
print(subset(vposts, vposts$odometer == 9500000))
# No hints: NA
vposts[which(vposts$odometer == 9500000 & !is.na(vposts$odometer)), ]$odometer = NA
plot(table(vposts$odometer), main = "Odometer With Four Max Removed", xlab = "Odometer",
        ylab = "Count")
# Satisfied with max values, now checking the year and condition for the zeros.
table(vposts$odometer == 0) # 89
vposts[ !is.na(vposts$odometer) & vposts$odometer == 0, c("year", "condition")]
# Most conditions look right, but a few used and most years not 2015
vposts[ !is.na(vposts$odometer) & vposts$odometer == 0 & vposts$year < 2015, c("body")]
# Only one body with info about odometer:
vposts[which((vposts$header == "1996 lexus") & (vposts$odometer == 0)), ]$odometer =
        220000
# Changing the rest to NA:
vposts[!is.na(vposts$odometer) & (vposts$odometer == 0) & (vposts$year < 2015), ]$odometer
        = NA
# Looking at odometers of 1:
table(vposts$odometer == 1)
vposts[ !is.na(vposts$odometer) & vposts$odometer == 1, c("year", "condition")]
 # Only 18 values (5 with conditions)
# Years range from 1967 to 2010 which doesn't makes sense for an odometer of zero
vposts[ !is.na(vposts$odometer) & vposts$odometer == 1, c("title", "body")]
vposts[which(vposts$title == "2002 Chevy Cavalier 4 Door-Automatic! Locally Owned with
        Clean CARFAX! - $1962 (castle rock)"), ]$odometer = 153175
vposts[which(vposts$title == "2006 KENWORTH T2000 - $22000 (SOUTH HOLLAND)"),
        ]$odometer = 1000
# Change rest to NA:
vposts[!is.na(vposts$odometer) & vposts$odometer == 1, ]$odometer = NA
plot(table(vposts$odometer), main = "Cleaned Odometer", xlab = "Odometer", ylab = "Count")
```

Question 2

Insight 1

```
vposts$timeUpdated = vposts$updated - vposts$posted
table(vposts$timeUpdated)
time = subset(vposts, !is.na(vposts$updated)) # In seconds
noUpdate = subset(vposts, is.na(vposts$updated)) # In seconds
par(mfrow = c(1,2))
plot(table(time$year), main = "Year of Posts That Were Updated", xlab = "Year", ylab =
        "Count", xlim = c(1921, 2016))
plot(table(noUpdate$year), main = "Years of Posts That Were Not Updated", xlab = "Year", ylab
        = "Count", xlim = c(1921, 2016))

# # # # # Insight 2
# odometer and city
par(mfrow = c(2,2))
compact = subset(vposts, vposts$size == "compact" & vposts$odometer < 100001)
hist(compact$odometer, main = "Odometer for Compact Vehicles", ylim = c(0, 350),
        breaks=c(0, 10000,20000,30000,40000,50000,60000,70000,80000,90000, 100000), xlim
        = c(0, 100000), xlab = "Odometer")
full = subset(vposts, vposts$size == "full-size" & vposts$odometer < 100001)
hist(full$odometer, main = "Odometer for Full-size Vehicles", ylim = c(0, 350), breaks=c(0,
        10000,20000,30000,40000,50000,60000,70000,80000,90000, 100000), xlim = c(0,
        100000), xlab = "Odometer")
mid = subset(vposts, vposts$size == "mid-size" & vposts$odometer < 100001)
hist(mid$odometer, main = "Odometer for Mid-size Vehicles", ylim = c(0, 350), breaks=c(0,
        10000,20000,30000,40000,50000,60000,70000,80000,90000, 100000),  xlim = c(0,
        100000), xlab = "Odometer")
sub = subset(vposts, vposts$size == "sub-compact" & vposts$odometer < 100001)
hist(sub$odometer, main = "Odometer for Sub-compact Vehicles", ylim = c(0, 350), breaks=c(0,
        10000,20000,30000,40000,50000,60000,70000,80000,90000, 100000),  xlim = c(0,
        100000), xlab = "Odometer")

# # # # # Insight 3
# City and price?
par(mfrow = c(1,1))
citySub = subset(vposts, vposts$price < 3e+05)
# Removing a few high values makes things much easier to see and since box plot uses median,
# and q1 and q3, removing a few high values doesn't change the graph much
plot(citySub$city, citySub$price, names = c("Boston", "Chicago", "Denver", "Las Vegas",
        "NYC", "Sac", "SF Bay"), main = "Price by City", xlab = "City", ylab = "Price")
```

References

Used piazza for some ideas and clarifications and used the ?help function in R.