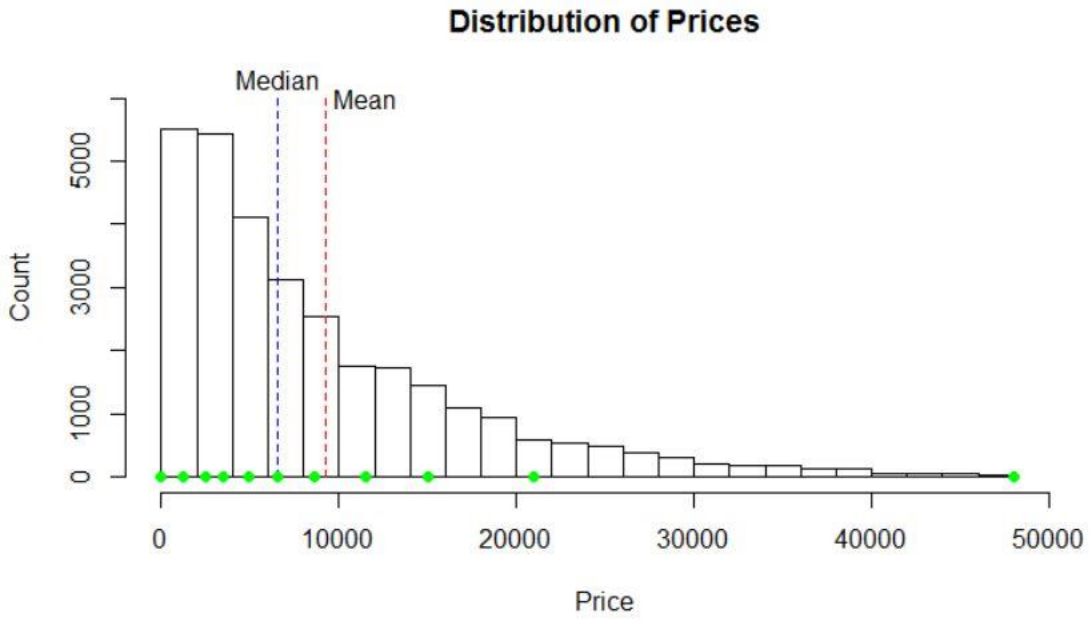
Analysis of Vehicle Sales Data

1. How many observations are there in the data set?
Answer: 34677

2. What are the names of the variables? And what is the class of each variable?

```
$id
[1] "character"

$updated
[1] "POSIXct" "POSIXt"

$cylinders
[1] "integer"

$time
[1] "POSIXct" "POSIXt"

$title
[1] "character"

$drive
[1] "factor"

$fuel
[1] "factor"

$description
[1] "character"

$body
[1] "character"

$odometer
[1] "integer"

$size
[1] "factor"

$location
[1] "character"

$lat
[1] "numeric"

$type
[1] "factor"

$transmission
[1] "factor"

$url
[1] "character"

$long
[1] "numeric"

$header
[1] "character"

$byOwner
[1] "logical"

$price
[1] "integer"

$posted
[1] "POSIXct" "POSIXt"

$condition
[1] "factor"

$city
[1] "factor"

$year
[1] "integer"

$maker
[1] "character"

$makerMethod
[1] "numeric"
```

3. What is the average price of all the vehicles? The median price? And the deciles? Displays these on a plot of the distribution of vehicle prices.



4. What are the different categories of vehicles? What is the proportion for each category?

type

bus	convertible	coupe	hatchback	mini-van	offroad	other	pickup
0.0006	0.0204	0.0469	0.0236	0.0131	0.0019	0.0192	0.0262

sedan	SUV	truck	van	wagon
0.2030	0.1214	0.0347	0.0146	0.0161

5. Display the relationship between fuel type and vehicle type. Does this depend on transmission type?

Fuel Type vs. Vehicle Type:

type	fuel: diesel	electric	gas	hybrid	other
bus	10	0	12	0	0
convertible	0	0	657	0	8
coupe	3	2	1518	1	33
hatchback	7	20	625	93	22
mini-van	0	0	411	1	3
offroad	1	0	65	0	0
other	24	0	307	2	333
pickup	109	0	774	0	22
sedan	52	9	6166	78	145
SUV	38	0	3495	28	171
truck	325	1	710	0	23
van	54	0	409	1	22
wagon	9	0	481	3	20

Now sorted by Transmission Type:

, , transmission = automatic

type	fuel: diesel	electric	gas	hybrid	other
bus	10	0	8	0	0
convertible	0	0	436	0	5
coupe	1	1	1046	1	18
hatchback	4	18	383	87	14
mini-van	0	0	408	1	3
offroad	1	0	41	0	0
other	13	0	227	1	269
pickup	96	0	654	0	20
sedan	39	4	5395	74	132
SUV	35	0	3157	27	148
truck	249	0	614	0	21
van	54	0	392	1	18
wagon	9	0	409	3	13

, , transmission = manual

type	fuel: diesel	electric	gas	hybrid	other
bus	0	0	4	0	0
convertible	0	0	197	0	2
coupe	2	1	421	0	8
hatchback	3	0	206	2	4
mini-van	0	0	1	0	0
offroad	0	0	24	0	0
other	10	0	41	0	8
pickup	13	0	114	0	2
sedan	9	0	498	1	1
SUV	1	0	129	0	0
truck	70	0	86	0	0
van	0	0	2	0	0
wagon	0	0	60	0	1

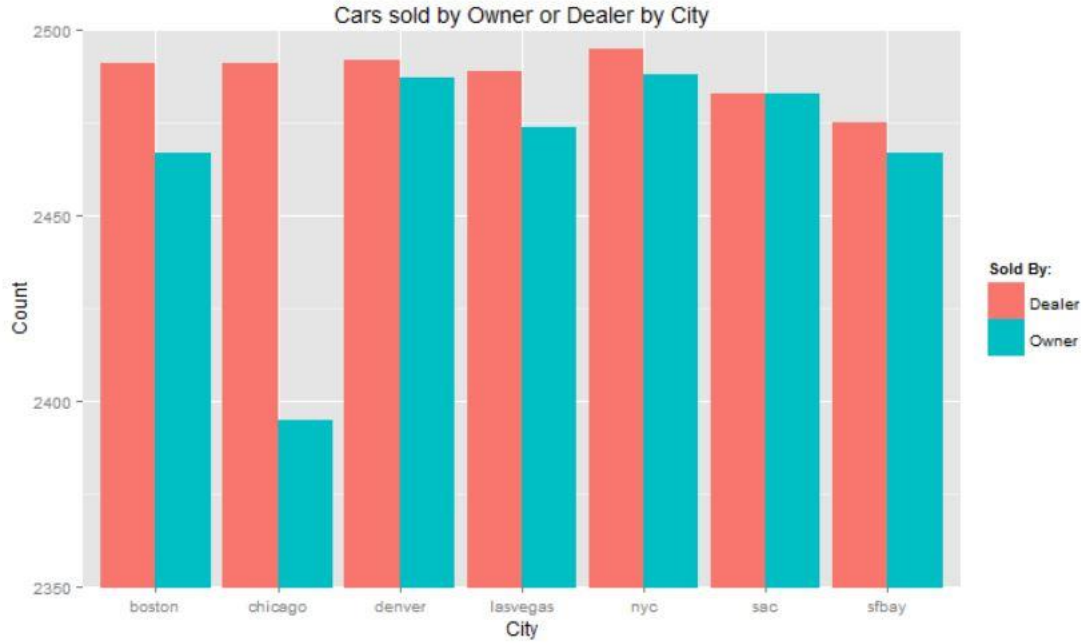
, , transmission = other

type	fuel: diesel	electric	gas	hybrid	other
bus	0	0	0	0	0
convertible	0	0	24	0	1
coupe	0	0	49	0	7
hatchback	0	2	34	3	3
mini-van	0	0	2	0	0
offroad	0	0	0	0	0
other	1	0	30	1	30
pickup	0	0	5	0	0
sedan	4	5	258	2	9
SUV	2	0	201	1	21
truck	6	1	10	0	2
van	0	0	15	0	4
wagon	0	0	12	0	6

6. How many different cities are represented in the dataset?

Answer: There are 7 different cities – Boston, Chicago, Denver, Las Vegas, NYC, Sacramento, and SF Bay

7. Visually display how the number/proportion of "for sale by owner" and "for sale by dealer" varies across city?



8. What is the largest price for a vehicle in this dataset? Examine and fix this value. Now examine the new highest value for price.
(See code for Question 3 where I remove the outliers before I graph it.)
Answer: Original max value: 600030000
New max value: 30002500

9. What are the three most common makes of cars in each city for "sale by owner" and "sale by dealer"? Are they quite similar or quite different?
Top 3 in each city sold by dealer:

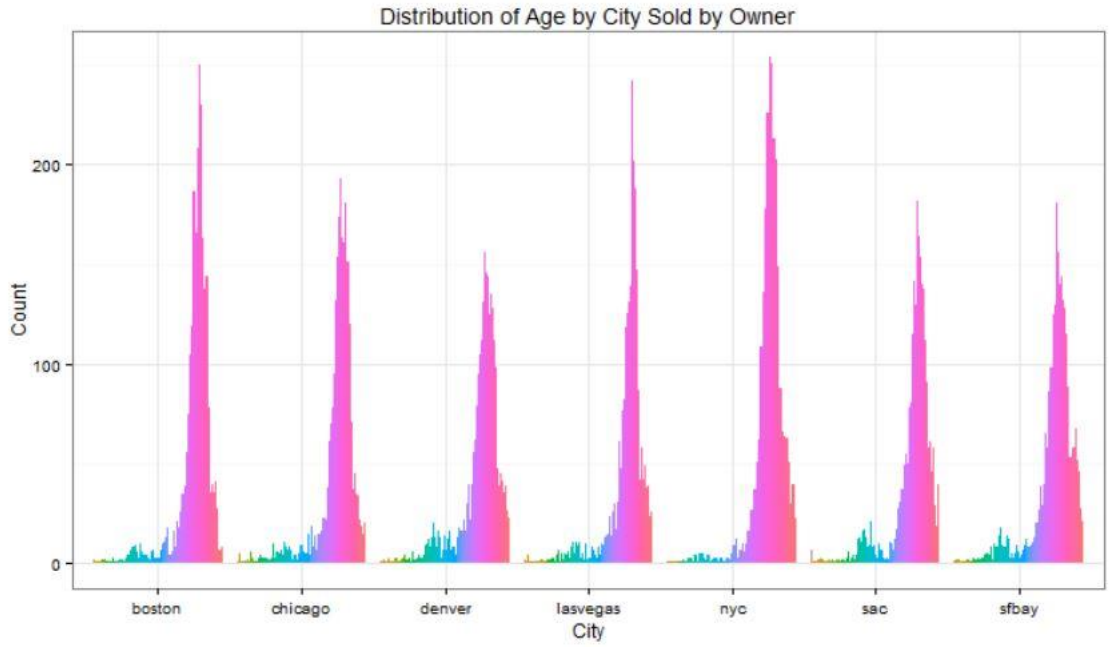
```
> top3_deal("boston")
    ford   toyota chevrolet
     333      288      215
> top3_deal("chicago")
chevrolet     ford   nissan
      305      305      208
> top3_deal("denver")
    ford chevrolet    dodge
     313      291      210
> top3_deal("lasvegas")
    ford   nissan chevrolet
     307      249      238
> top3_deal("nyc")
nissan toyota  honda
   328    238    220
> top3_deal("sac")
    ford   toyota chevrolet
     337      273      206
> top3_deal("sfbay")
toyota   ford    bmw
   269    245    227
```

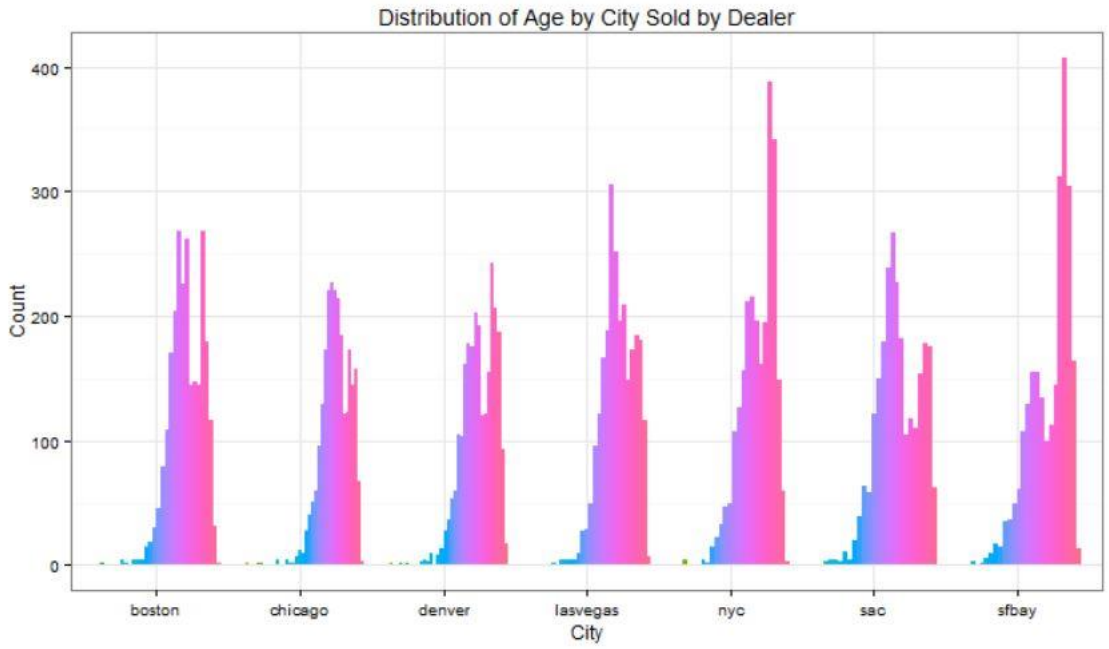
Top 3 in each city sold by owner:

```
> top3_own("boston")
     ford     honda chevrolet
      353       263       226
> top3_own("chicago")
chevrolet      ford     honda
      365       331       180
> top3_own("denver")
     ford chevrolet    toyota
      378       313       191
> top3_own("lasvegas")
     ford chevrolet    toyota
      394       306       193
> top3_own("nyc")
nissan toyota  honda
   308    274    260
> top3_own("sac")
    toyota      ford chevrolet
       340       305       299
> top3_own("sfbay")
toyota  honda   ford
   332    322    257
```

In both categories of "sold by dealer" and "sold by owner", Ford is in almost every top 3. Similarly, Toyota, Chevrolet, and Honda show up in the top 3 of many cities as well. They are quite similar, and this makes sense logically since the top brands are very well-known across the nation.

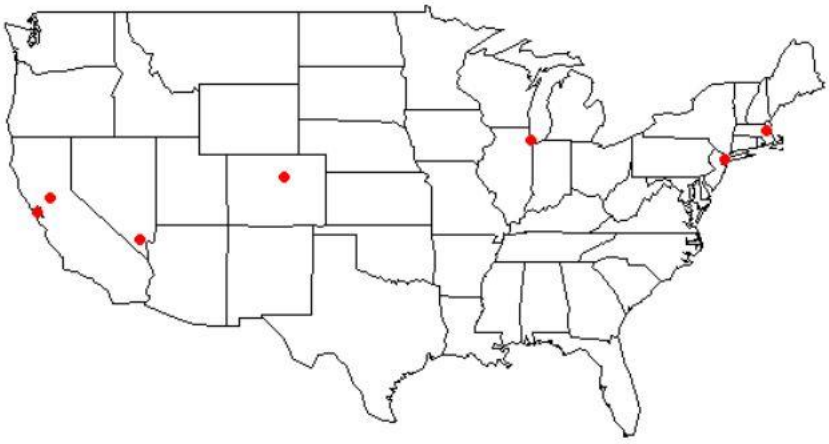
10. Visually compare the distribution of the age of cars for different cities and for "sale by owner" and "sale by dealer". Provide an interpretation of the plots, what are the key conclusions and insights?





In the plots above, the years of the vehicles are distributed by city and also by seller. When the vehicles are sold by the owners, it looks like the majority of the vehicles are sold are significantly older than the ones sold by dealers. According to the graphs, the dealers are also selling vehicles in a wider range of years than the owners. Overall, dealers sell more cars than owners do.

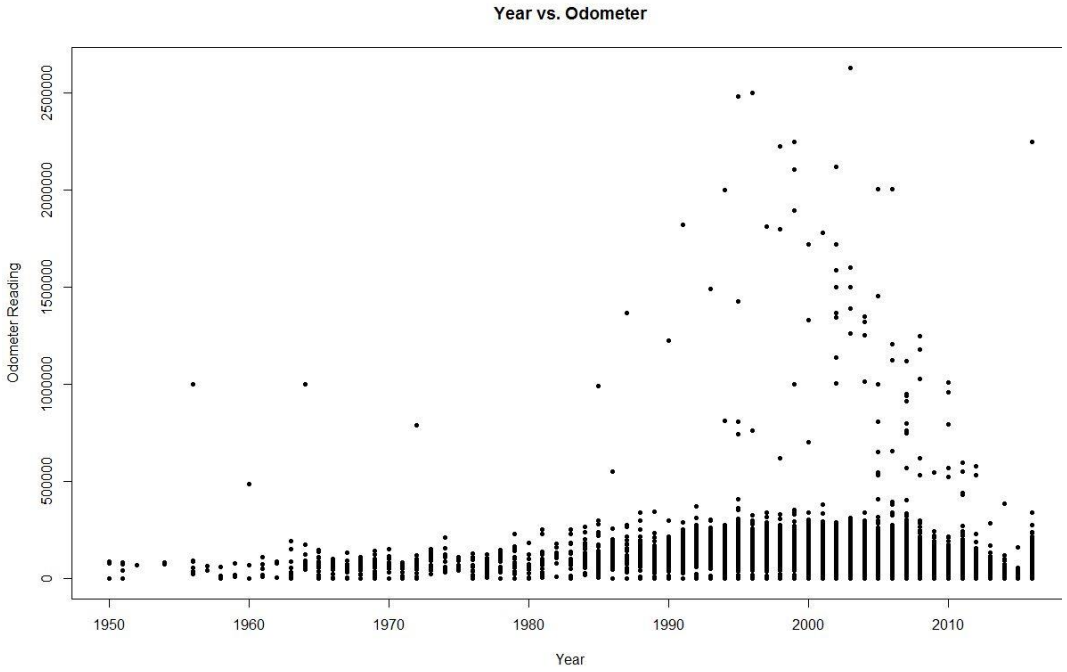
11. Plot the locations on a map. What do you notice?



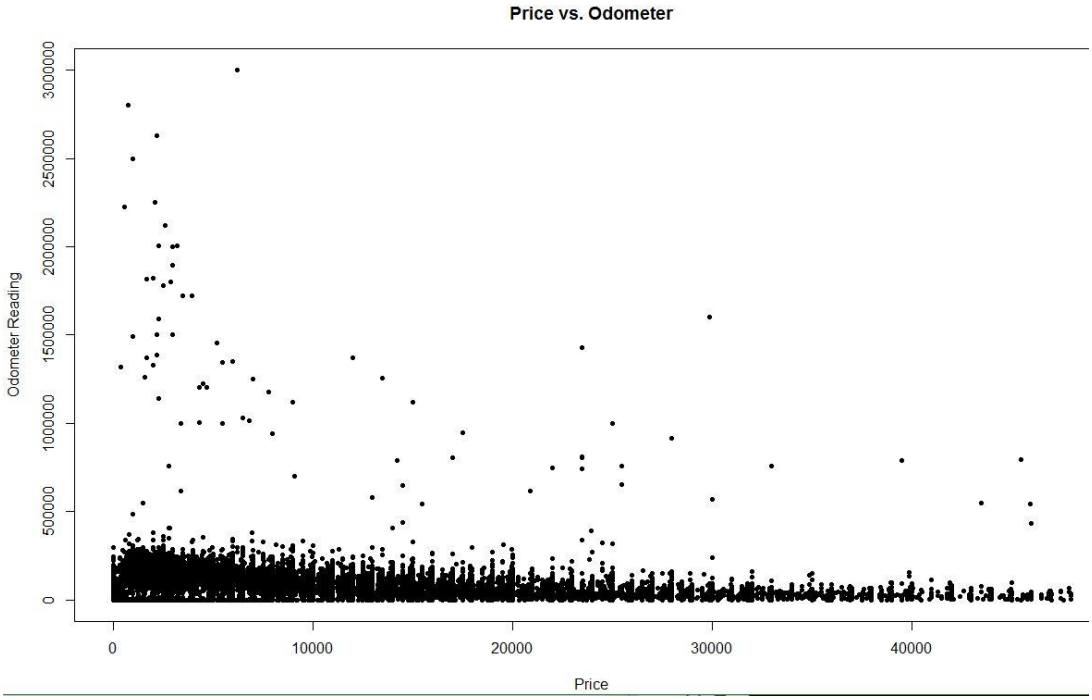
As shown on the map, the locations are in high-density cities where populations are also high. It is logical since with many people comes more cars as well as more cars being bought and sold. Sacramento and the SF Bay are also very close to each other as are NYC and Boston. The other three cities are pretty far from any other cities.

12. Summarize the distribution of fuel type, drive, transmission, and vehicle type.

13. Plot odometer reading and age of car? Is there a relationship? Similarly, plot odometer reading and price? Interpret the result(s). Are price and age of car related?



There is a relationship between odometer reading and age of car. As the year of the car increases, so does the odometer rating in general. This could be because since the car is newer, there have been less number of miles driven in it. Therefore, the result would be a higher odometer reading.



There is a relationship between price and odometer reading, however it seems to be the opposite of the relationship between odometer and year. The higher the odometer reading, the less pricey the car is. This could mean that people are only willing to pay so much for a certain odometer reading.

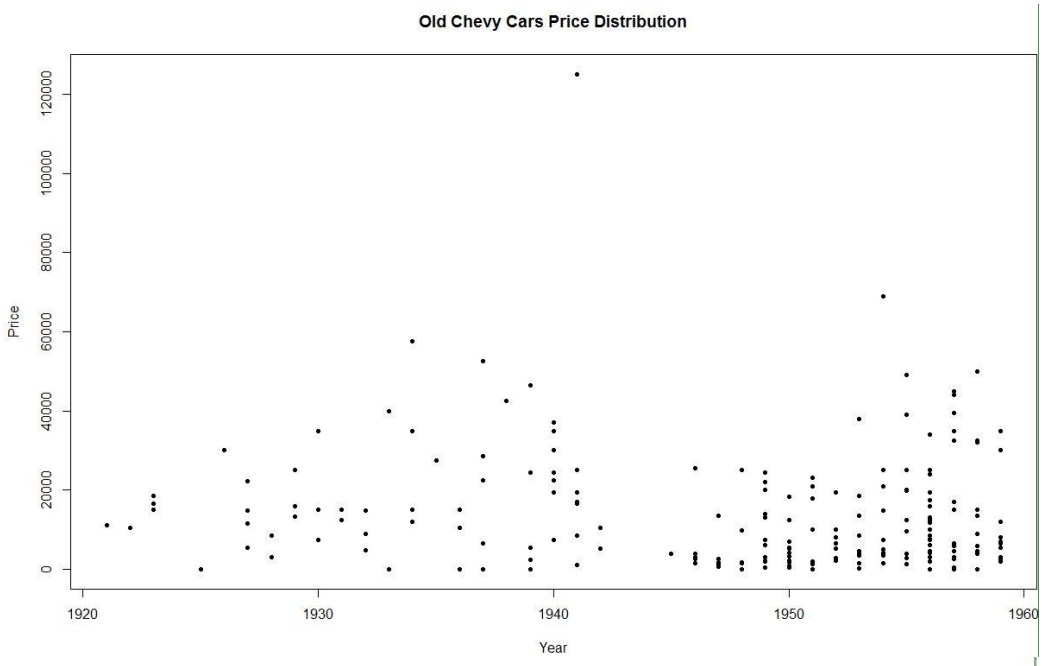
By looking at the individual relationships, it seems plausible that there is a relationship between age of the car and price. The older the car, the lower the price. Logcially, this makes sense because the older the car, the more miles it has, and the lower the price people are willing to pay for such a car.

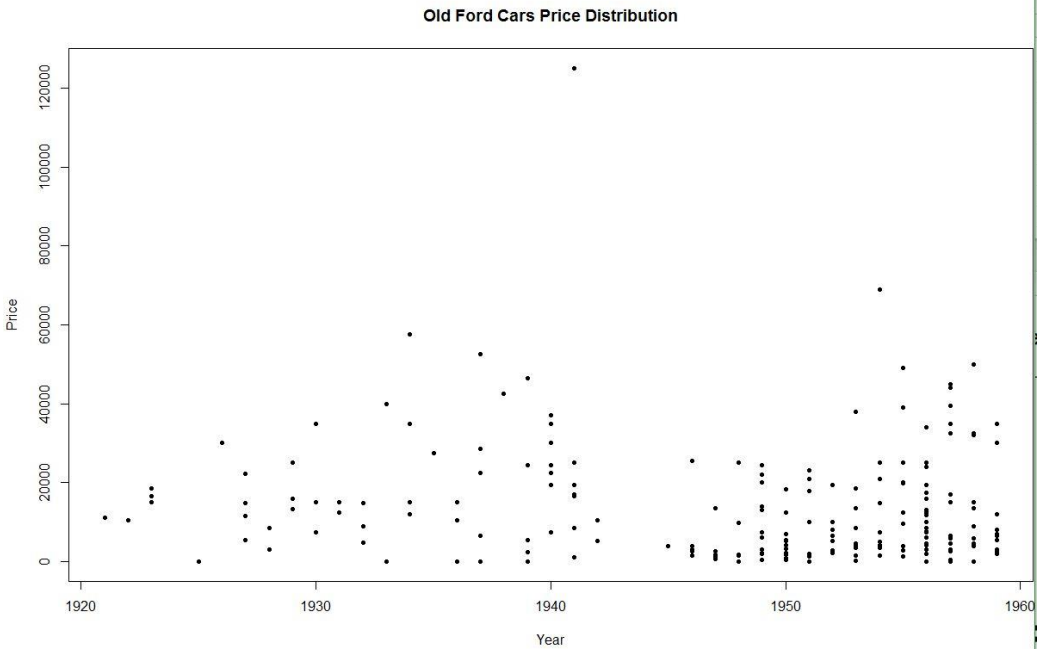
14. Identify the "old" cars. What manufacturers made these? What is the price distribution for these?

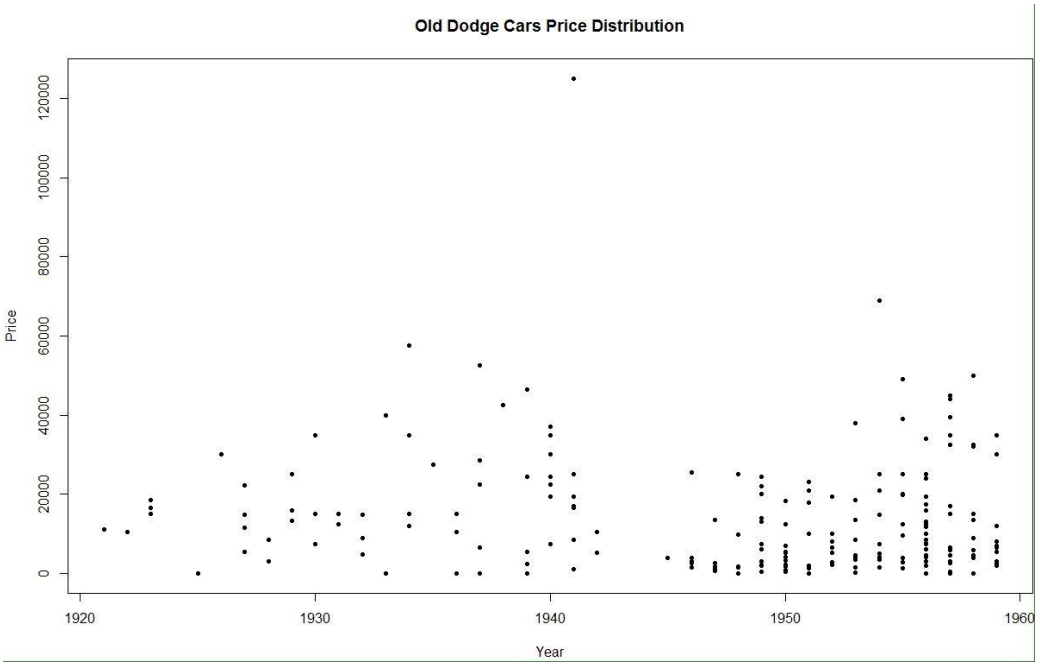
“Old” Cars and the manufacturers that made them:

ford	chevrolet	NA's	dodge	buick	willys	mercury
84	66	13	12	10	9	7
plymouth	cadillac	lincoln	pontiac	chrysler	gmc	hudson
6	5	5	5	4	4	4
oldsmobile	international	mg	bugatti	jeep	studebaker	desoto
4	3	3	2	2	2	1
mercedes	rolls royce	volkswagen				
1	1	1				

Price Distribution graphs by Top 3 “Old” Car manufacturers:


Old Chevy Cars Price Distribution
Price
0
20000
40000
60000
80000
100000
120000
1920
1930
1940
1950
1960
Year

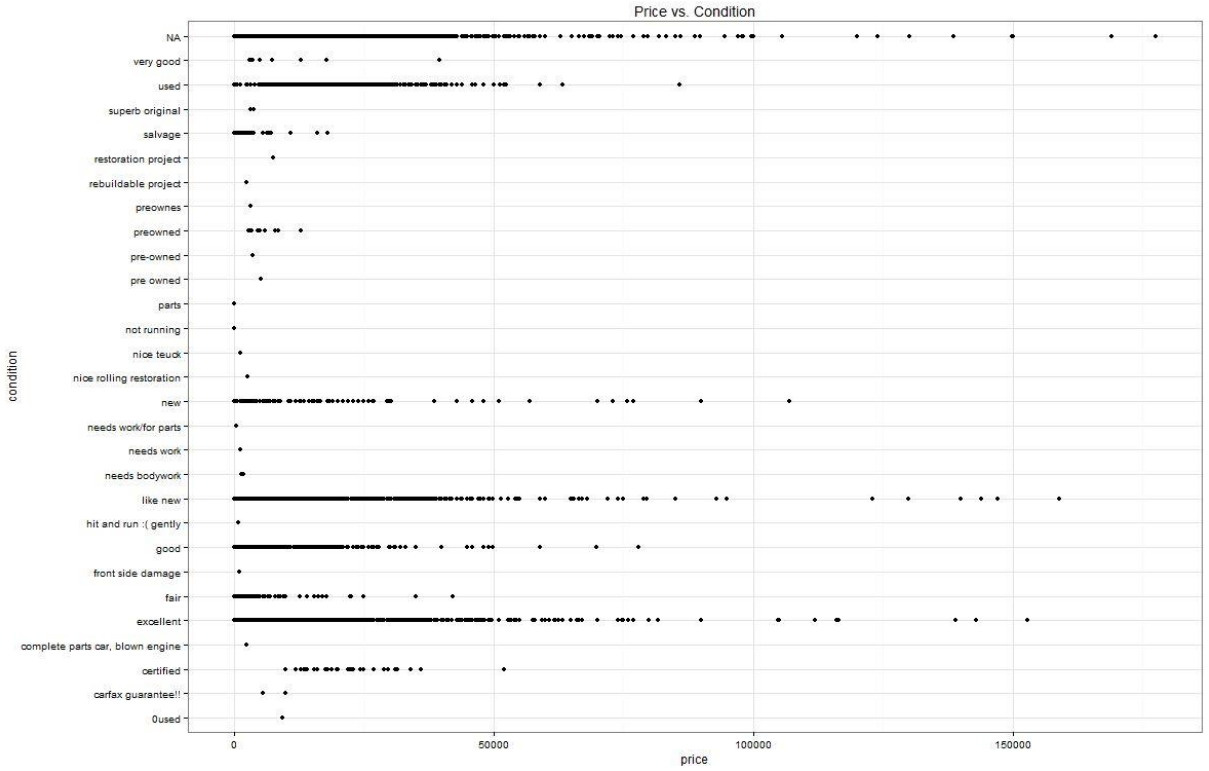

Old Ford Cars Price Distribution
Price
0
20000
40000
60000
80000
100000
120000
1920
1930
1940
1950
1960
Year

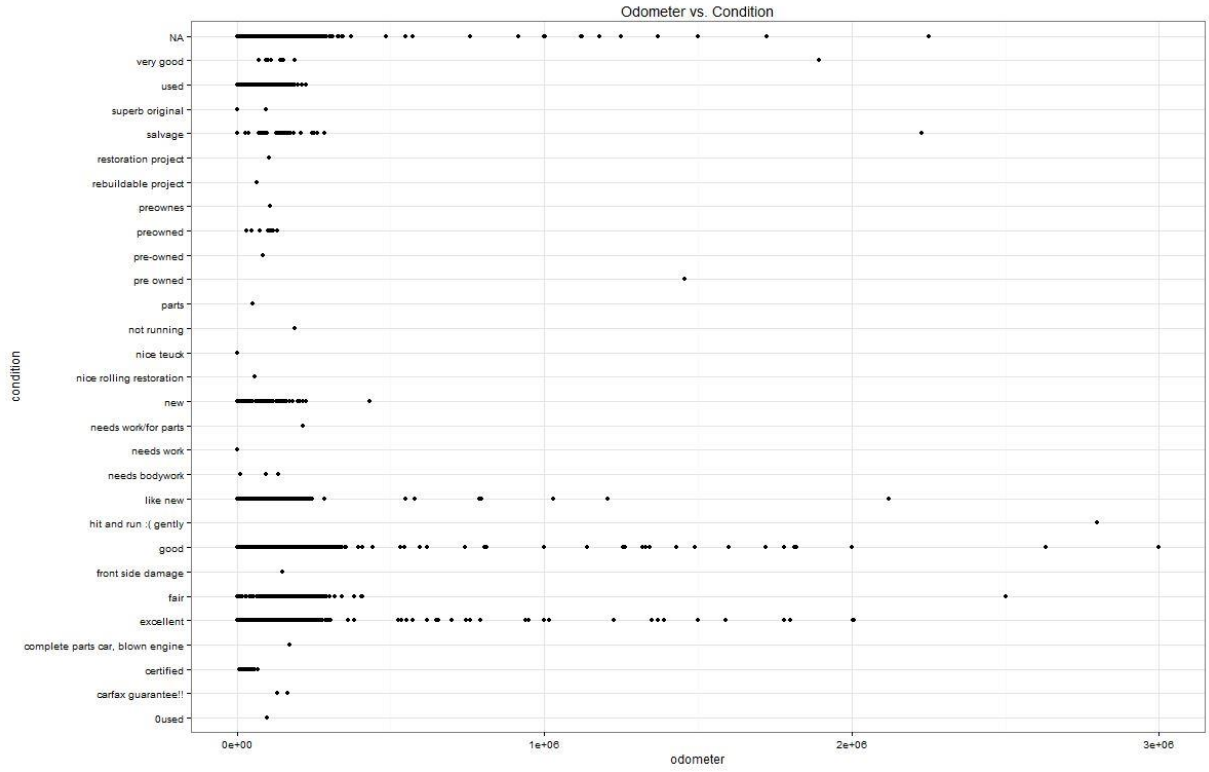


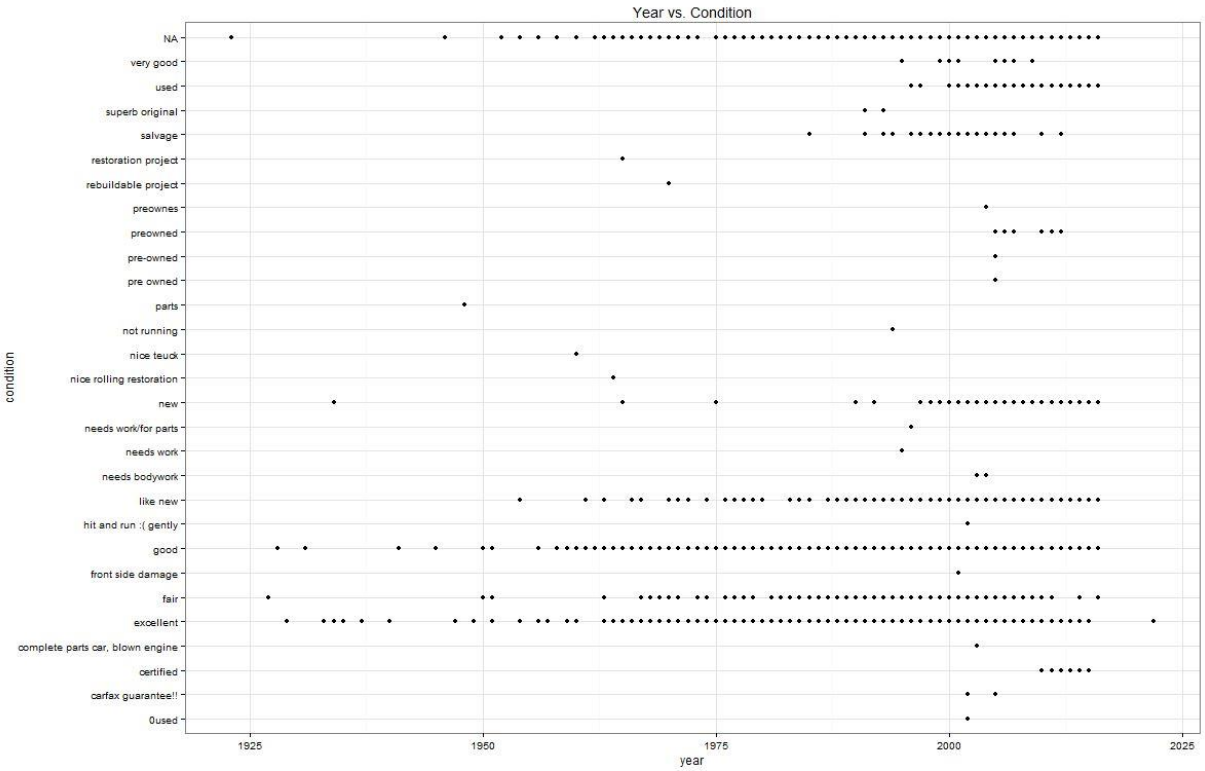
15. I have omitted one important variable in this data set. What do you think it is? Can we derive this from the other variables? If so, sketch possible ideas as to how we would compute this variable.

I think the omitted variable is MPG, or miles per gallon. We can possible derive this from the odometer reading, however, that would be very complicated. Another method would be to simply extract the information from the body of the post. We could do this by subsetting the dataset into just the column with the body and then read the information in it and parce it to find what we are looking for.

16. Display how condition and odometer are related. Also how condition and price are related. And condition and age of the car. Provide a brief interpretation of what you find.


Price vs. Condition
condition
NA
very good
used
superb original
salvage
restoration project
rebuildable project
preownes
preowned
pre-owned
pre owned
parts
not running
nice teuck
nice rolling restoration
new
needs work/for parts
needs work
needs bodywork
like new
hit and run :(gently
good
front side damage
fair
excellent
complete parts car, blown engine
certified
carfax guarantee!!
0used
0
50000
100000
150000
price


Odometer vs. Condition
condition
NA
very good
used
superb original
salvage
restoration project
rebuildable project
preownes
preowned
pre-owned
pre owned
parts
not running
nice teuck
nice rolling restoration
new
needs work/for parts
needs work
needs bodywork
like new
hit and run :(gently
good
front side damage
fair
excellent
complete parts car, blown engine
certified
carfax guarantee!!
0used
0e+00
1e+06
2e+06
3e+06
odometer



Looking at all three graphs, it is apparent that the label “pre-owned” is not a good indicator of the car’s price, mileage, or age. There are many labels for the condition of the cars, however, it is very difficult to group them together in order to condense them. As expected, the higher the price, the more recent the year, and the lower the odometer reading, the better condition the cars are in.

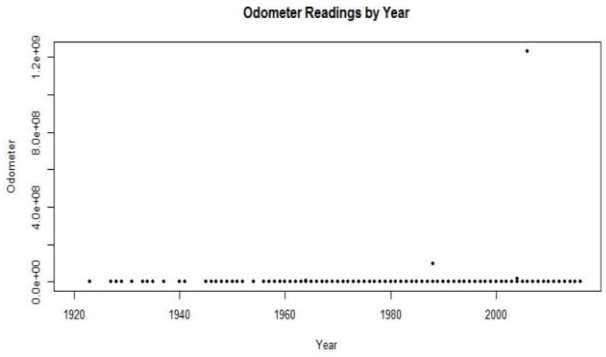
~~~~~~

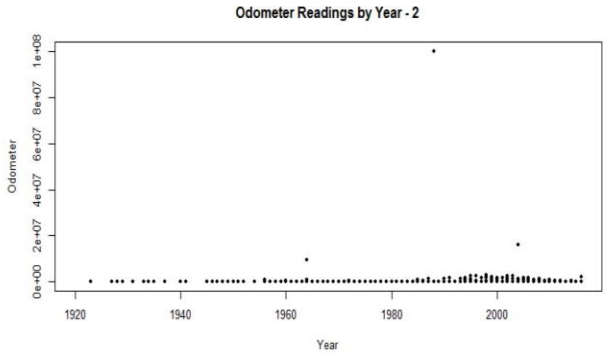
Finding Anomalies, Cleaning Data, and Interesting Insights

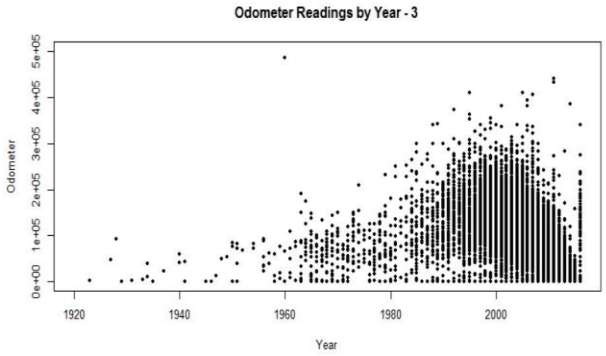
Anomalies

1. Odometer Readings

Firstly, there are some very high odometer readings for many cars in the dataset. Readings above 1 million miles are not as uncommon in this dataset as expected. That being said, it is still possible for a car to reach 1 million miles, but it is improbable that this many cars with such a reading are being sold within the same month. Below, I have graphed the odometer readings against the year of the car 3 different times. The only thing that I changed is the range of the y-axis.
~~~~~~







Then, I also looked at the low odometer readings. I noticed that there are many cars with odometer readings of zero. This could be a marketing or search filter strategy on behalf of the sellers. Out of curiosity, I made a table of all the zero odometer readings by city, as shown below. Interestingly, Denver has a lot of cars with zero odometer readings. I don't know why, but I thought it was weird.

boston	chicago	denver	lasvegas	nyc	sac	sfbay
4	13	38	8	15	6	5

Personally, I would consider both the extremely large and small odometer readings to be outliers, and therefore, I would find the IQR and calculate the boundaries that way to ensure consistency. However, there are many ways to define "old" cars, so the data can be subsetted in many ways.

2. Time vs. Updated Time

In some postings, the time that the post was posted is later than the time that the post was updated. Logically, this doesn't make sense. According to my analysis, there are 13,209 postings that have this characteristic. But since there are so many, there must be another reason why this is so. My thinking is that these posts were re-uploaded to the site because the cars were not being sold. That being said, I counted the instances of this by city, and the table is shown below.

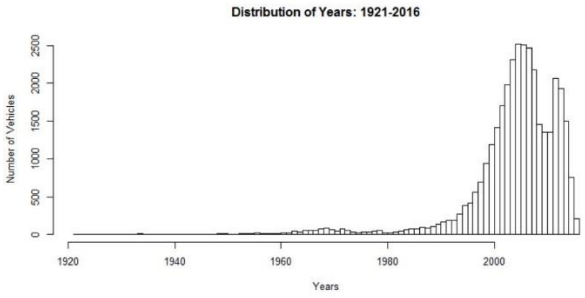
boston	chicago	denver	lasvegas	nyc	sac	sfbay
2724	2366	2030	1290	2390	1240	1169

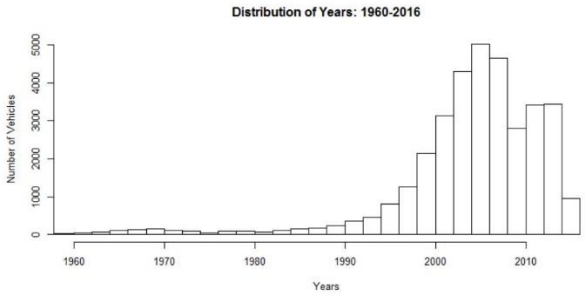
Maybe in cities like Boston, Chicago, Denver, and NYC, the sellers have trouble selling their cars quickly, so they re-upload them to have them show up first in the sequence. I wouldn't remove any of these from the overall data because of two reasons: 1) there is so many of them that it would most definitely affect any sort of distribution and 2) there is still other valuable data in these postings.

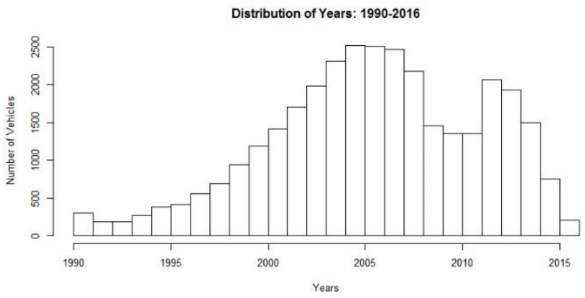
3. Year

There are two major outliers where the years are 4 and 2022. Upon closer inspection, neither of these posts have the actual years of the car in the title or the description. Therefore, we can discard these two specific points as long as we are just focusing on the years. The second lowest year vehicles are all from the year 1900. All seven of these cars are from the greater Sacramento area, and all but one were posted by a company who was asking to buy cars that did not pass smog checks. They are offering to pay $750. I imagine that these are not the only posts that are for people wanting to buy vehicles rather than sell them. However, in this case, it would be easiest to just remove these.

Below is the distribution of years of cars shown three ways. Each graph has a different range of years. As seen in these graphs, there is a lot of "old" cars that skew the histogram and make it hard to read.



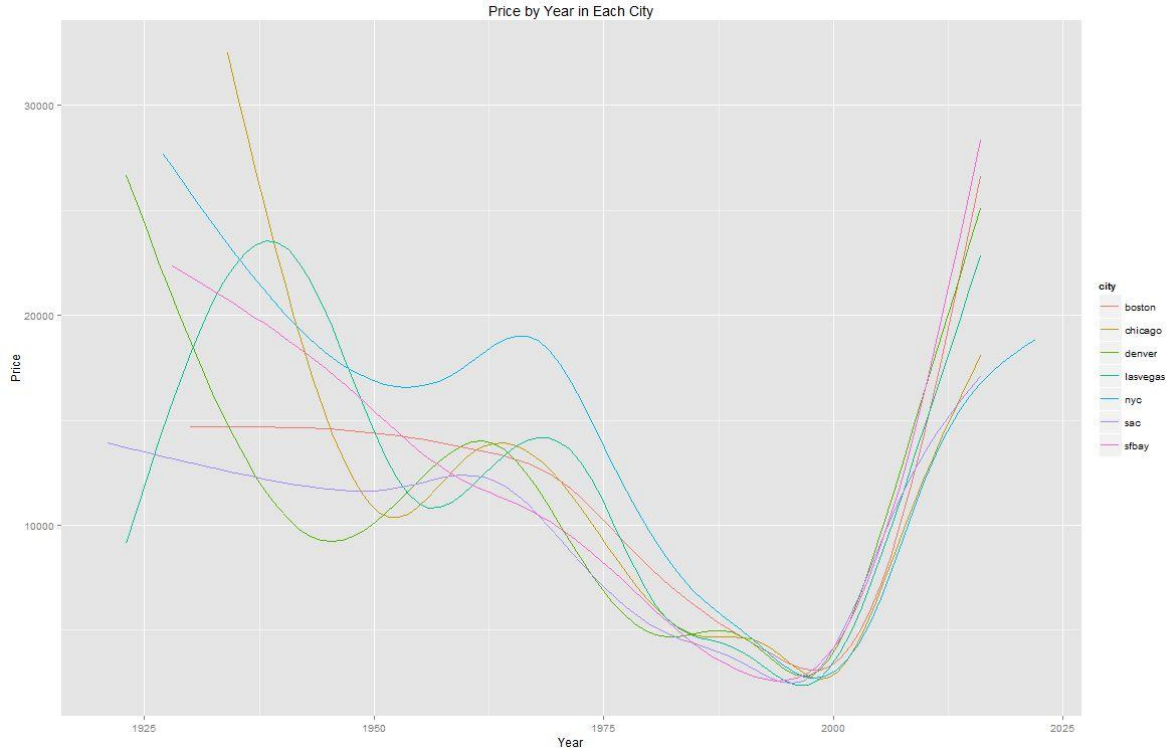




Interesting Insights

1. Price by Year in Each City

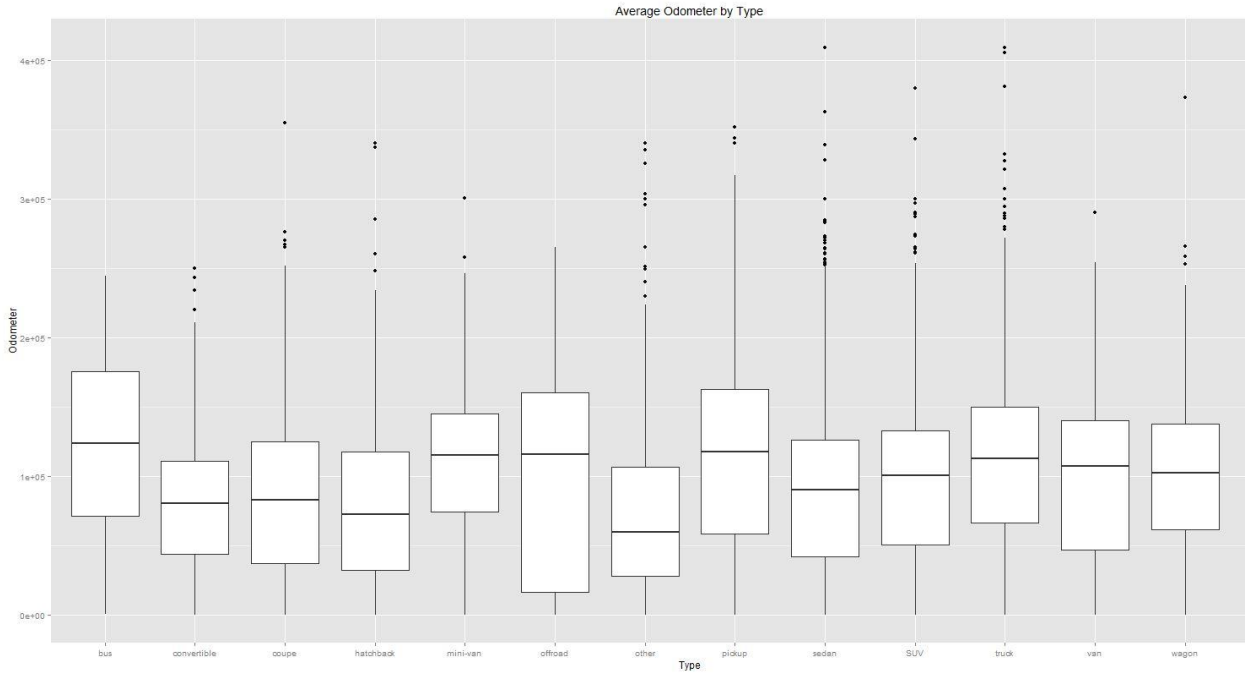
Using the ggplot2 library, I was able to plot the price of the vehicles in each city by year. I found a way to plot a smooth conditional mean, which automatically uses a generalized additive model function instead of the default local polynomial regression fitting (which is used for under 1,000 data points). Since this data set is so large, this is the best way I found. Here is the graph below:



According to the graph above, there are some basic trends apparent in all the cities. For example, prices in all cities drop drastically around vehicles from the 2000s. I do not know the exact reason for this, but it could be that those cars are too young to be considered "vintage", but too old to be considered "recent". There is no one city that has consistently higher prices than the rest, although I would have expected NYC or SF Bay to have higher prices.

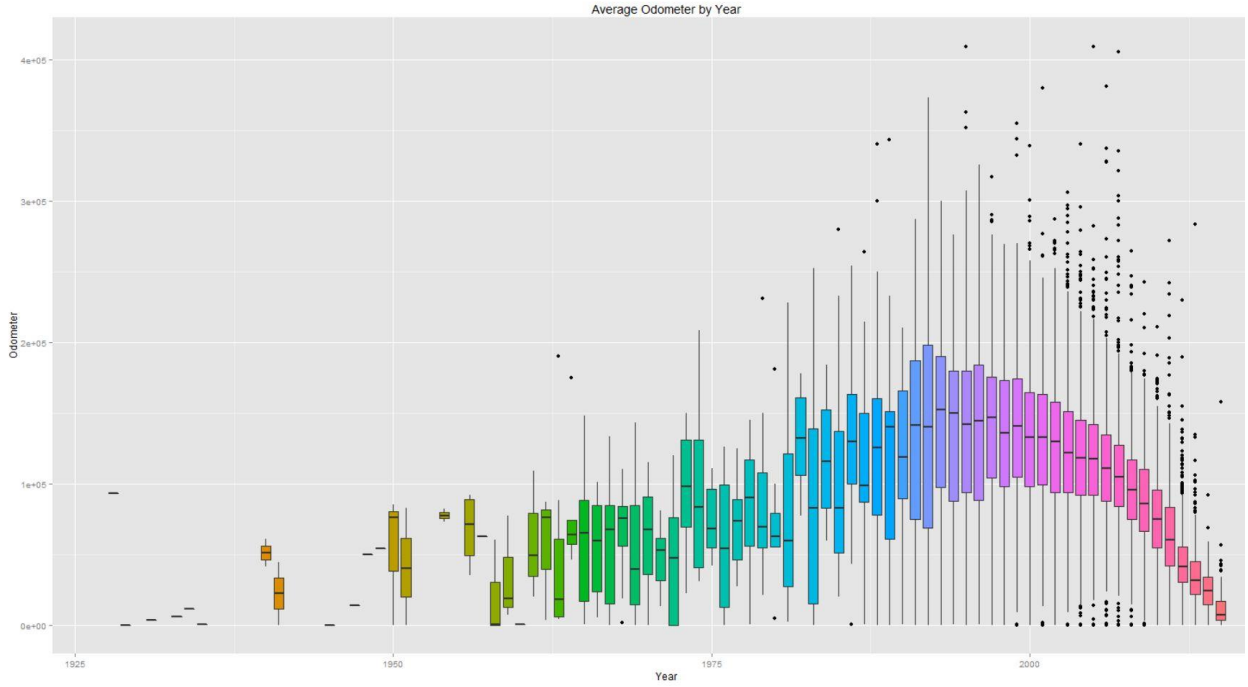
2. Average Odometer by Type and by Year

Another interesting insight I found was that the average odometer reading by type:



As the graph above shows, some types of vehicles have higher average odometer readings than others. Examples of these are buses, trucks, pickups, SUVs, and mini-vans. Some types like convertibles, coupes, and hatchbacks have lower averages, which makes sense. These types of vehicles are not meant for long distances, so their owners probably did not take them on super long drives.

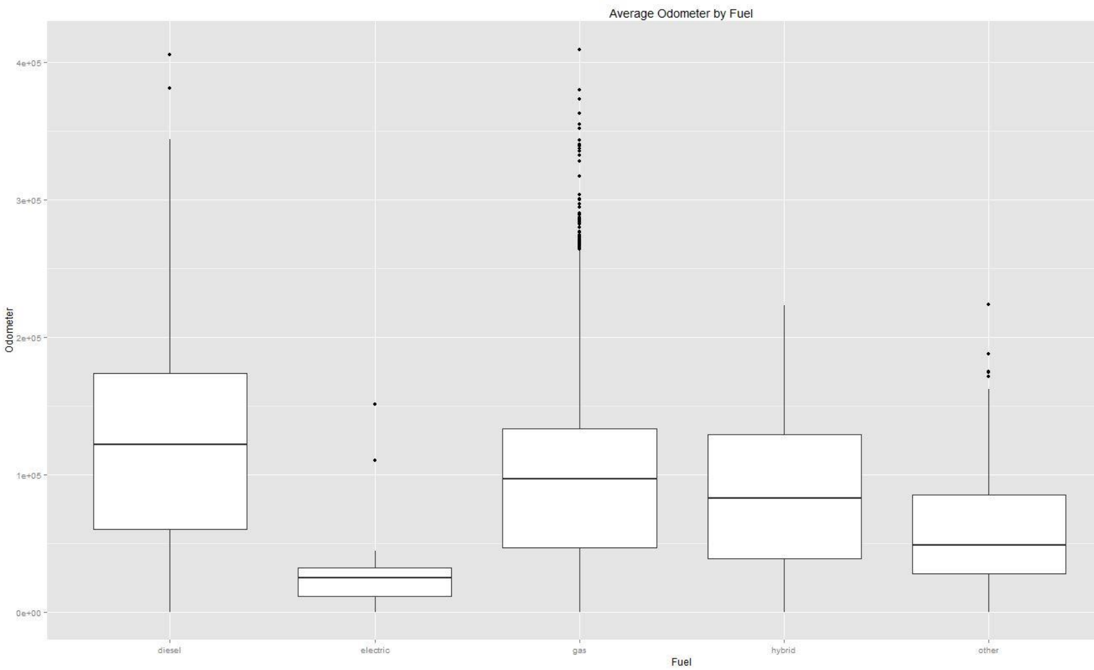
I also decided to look at the average odometer reading per year of car:



This graph is super interesting because it looks like the older cars that are being sold have lower averages than the newer cars. It probably means that people in those years did not drive as much as we do nowadays. As cities have expanded, it has become more necessary for people to travel farther distances every day, therefore accumulating more miles on their vehicles.

3. Average Odometer by Fuel Type

I graphed the average odometer readings by fuel type and here is graph below:



It is very interesting that electric vehicles by far have a lower average odometer reading than other vehicles with different fuel types. Strangely, diesel vehicles also have higher average odometer readings than gasoline vehicles overall. I would have guessed, that electric cars may have more miles on them because they are so efficient and light, however, that does not seem to be the case.

(Side note: With the dataset created for this, I tried to omit all the rows that had NA values in them. When I did so and plotted the same graph as above, all the electric cars were gone. But then I realized that all electric car engines don’t have any cylinders, so that column would have an NA value for sure.)

This information may be useful to electric car companies as it may be a flaw in their business plans. People may not trust their electric cars to travel long distances like their gasoline cars. Therefore, they would not accumulate many miles on their electric cars. At the same time, this increases their resale value, so that could be a helpful marketing feature.

~~~~~~
~~~~~~

Text Processing with Pattern Matching & Regular Expressions

1. I extracted the price from the body column, and then compared it to the numbers in the price column to see if they were the same. If there was no price found in the body, then I just made the value equal to zero. Approximately **91.15%** of the prices I extracted matched that of the price column.

2. I was able to extract the VIN number from the body column as well. With my regular expression, there were **5,169** posts with VIN numbers in them.

3. I found that there were **15,617** posts with phone numbers in them.

4. There were **103** posts with emails in them.

5. When I found the year in the body column, I got a 91.88% success rate. This means that I had **31,863** matches. But, when I found the year in the description column, the success rate jumped up to 99.69%, which equates to **34,572** matches.

6. Using the title column, I was able to find the model for 24,685 posts. Here is a snapshot of the first few models:

```
> head(numposts5)
[1] "Camaro"  "Equinox" "Altima"  "M35x"    "G37x"    "MDX"
```

Modeling

To suggest the approximate price for a car given its age, mileage, and condition, I decided to use the linear modeling method. I believe this is the best choice because it can also show which variables are actually relevant to the model and add value. Although k-nearest neighbors is a good modeling method, for this particular situation, it wouldn't have worked because there are many variables and we don't know which ones really show correlation to the price. Similarly, a regression tree would have been inappropriate in this situation since it doesn't do a very good job of using multiple variables to predict something. If we had only one or two variables, maybe it would have been better. Furthermore, there are obviously outliers in this dataset, so that may have messed with the results a little.

TOYOTA

I first looked at a model with all the variables included: odometer, year, condition, and city. The plot of this model looked pretty intense, but I couldn't make out any correlation between the predictors. I then looked at the summary of this model which showed that at a significance level of 0.01 and below, only the year, intercept, and city were relevant. None of the conditions were significant, which led me to completely throw out the condition predictor in my next model.

My second model had just three predictors: odometer, year, and city. It performed much better than the earlier model, and it showed that the cities of Denver, Las Vegas, Sacramento, and SF Bay had a high correlation to the price. This means that the prices of Toyotas sold in

these cities were either higher or lower because of their particular location. (Maybe Chicago and NYC have too many Toyotas being sold there, so the prices cannot be affected much because of the large supply.)

Just to be sure, I made a third model that had the variables: odometer, year, and condition. With the city excluded, this model performed very poorly. The results were very similar to the first model, and still none of the conditions were significant. With these results, I can best predict the price of a Toyota with just the mileage, age, and location.

BMW

I did the exact same thing with the BMWs as I did with the Toyotas, but the results were quite different. The odometer, year, and city were still important predictors, but condition became a very significant variable.

I made the same three models as before: first with all 4 predictors, second without condition, third without city. “Like new”, “new”, and “used” BMWs seem to be very correlated to price, meaning that if they have one of these conditions, then their price is either higher or lower because of it. Interestingly, NYC and SF Bay were the only two cities that showed a high correlation to the price. It might make sense because they are high density areas and people tend to have higher incomes in these places. In my opinion, the best way to predict the price of a BMW is to use all 4 variables.

Code Appendix

Part 1:

```
#==========Q1==========
dim(vposts)

#==========Q2==========
names(vposts)
sapply(vposts, class)

#==========Q3==========
price <- vposts$price[!is.na(vposts$price)]
quantile(price, prob = seq(0,1, length = 11))
quantile(price, prob = seq(0.9,1, length = 11))

price2 <- price[price <= 47997]
range(price2)
quantiles <- quantile(price2, prob = seq(0,1, length = 11))
mp2 = mean(price2)
medp2 = median(price2)
hist(price2, main = "Distribution of Prices", xlab = "Price", ylab = "Count",
    breaks = 20, xlim = c(0,50000), ylim = c(0,6000))

lines(c(mp2,mp2),c(0,6000), col = "red", lty = "dashed")
text(11500, 6000, "Mean")
lines(c(medp2,medp2),c(0,6000), col = "blue", lty = "dashed")
text(medp2, 6300, "Median")

points(quantiles, rep(0, 11), pch = 16, col = "green")

#==========Q4==========
levels(vposts$type)
type <- vposts$type
round(table(type)/length(type), digits = 4)

#==========Q5==========
#Relationship between Type and Fuel only
type_fuel = vposts[ ,c("type", "fuel")]
table(type_fuel)

#Relationship between Type and Fuel but sorted by Transmission
type_fuel_byTrans = vposts[ ,c("type", "fuel", "transmission")]
table(type_fuel_byTrans)

#==========Q6==========
levels(vposts$city)
```

```
#============Q7=============
library(ggplot2)

q7 <- (ggplot(vposts) + geom_bar(aes(x = city, fill = byOwner), position = "dodge")
+ coord_cartesian(ylim = c(2350,2500))
+ labs(title = "Cars sold by Owner or Dealer by City", x = "City", y = "Count")
+ scale_fill_discrete(name="Sold By:",
                       labels=c("Dealer", "Owner")))

#============Q8=============
#See code for Q3 where I remove the outlier before I graph it.
max_price = max(vposts$price, na.rm=TRUE)
p_no_max <- vposts$price[vposts$price < max_price & !is.na(vposts$price)]
max(p_no_max)

#============Q9=============
subdata9 <- vposts[c("city", "byOwner", "maker")]
subdata9_own <- subset(subdata9, byOwner==TRUE)
subdata9_deal <- subset(subdata9, byOwner==FALSE)

top3_own <- function(tempcity="boston"){
  temp <- subset(subdata9_own[c("city","maker")], city==tempcity)
  tempvec <- sort(summary(factor(temp$maker)), decreasing=T)
  tempvec[1:3]
}

top3_own("boston")
top3_own("chicago")
top3_own("denver")
top3_own("lasvegas")
top3_own("nyc")
top3_own("sac")
top3_own("sfbay")

top3_deal <- function(tempcity="boston"){
  temp <- subset(subdata9_deal[c("city","maker")], city==tempcity)
  tempvec <- sort(summary(factor(temp$maker)), decreasing=T)
  tempvec[1:3]
}

top3_deal("boston")
top3_deal("chicago")
top3_deal("denver")
top3_deal("lasvegas")
top3_deal("nyc")
```

```
top3_deal("sac")
top3_deal("sfbay")

#==========Q11==========
#Code found online at:
#http://www.r-bloggers.com/r-beginners-plotting-locations-on-to-a-world-map/
#I adapted it to fit the homework assignment though!
library("ggmap")
library(maptools)
library(maps)

visited <- c("Boston", "Chicago", "Denver", "Las Vegas", "NYC", "Sacramento", "San
Francisco")
ll.visited <- geocode(visited)
visit.x <- ll.visited$lon
visit.y <- ll.visited$lat

map("state")
points(visit.x,visit.y, col="red", pch=16)

#==========Q12==========
subdata12 <- subset(vposts[c("fuel","drive","transmission","type")])
q12 <- (ggplot(subdata12)
      + geom_bar(aes(x=transmission, fill=fuel),position="dodge")
      + facet_grid(type~drive, scales="free")
      + labs(title = "Transmission, Drive, Type, and Fuel"))

#==========Q13==========
#Year vs. Odometer
subdata13 <- vposts[ ,c("year","odometer")]
unique(vposts$year)
range(vposts$odometer, na.rm=TRUE)
subdata13 <- subset(subdata13, year >= 1950 & odometer < 9500000)
max(subdata13)
plot(subdata13, main = "Year vs. Odometer", xlab = "Year", ylab = "Odometer Reading",
   xlim = c(1950,2016), pch = 20)

subdata13_2 <- vposts[ ,c("price","odometer")]
subdata13_2 <- subset(subdata13_2, price <= 47997 & odometer < 9500000)
plot(subdata13_2, main = "Price vs. Odometer", xlab = "Price",
   ylab = "Odometer Reading", pch = 20)

#==========Q14==========
a <- rle(sort(vposts$year))
b <- data.frame(year=a$values, n=a$lengths)
q14_1 <- ggplot(b) + geom_line(aes(x=year,y=n)) + xlim(c(1920,2016))
```

```
#I chose the date range of 1920-1960 based on this graph

subdata14 <- subset(vposts[c("year","maker","price")], year>=1920 & year<1960)
sort(summary(factor(subdata14$maker)), decreasing=T)

subdata14.1 <- subset(subdata14[c("year", "price")], maker = "ford")
plot(subdata14.1, main ="Old Ford Cars Price Distribution", xlab = "Year", ylab = "Price",
    pch = 20)
subdata14.2 <- subset(subdata14[c("year", "price")], maker = "chevrolet")
plot(subdata14.2, main ="Old Chevy Cars Price Distribution", xlab = "Year", ylab = "Price",
    pch = 20)
subdata14.3 <- subset(subdata14[c("year", "price")], maker = "dodge")
plot(subdata14.3, main ="Old Dodge Cars Price Distribution", xlab = "Year", ylab = "Price",
    pch = 20)

#==========Q15==========
subdata15 <- vposts$body
subdata15[1:10]

subdata15.1 <- vposts$description
subdata15.1[1:10]

#==========Q16==========
subdata16 <- subset(vposts[c("odometer", "condition", "price", "year")], odometer<=5000000 &
price <=200000)
q16_1 <- (ggplot(subdata16)
        + geom_point(aes(y=condition, x=odometer))
        + theme_bw()
        + labs(title = "Odometer vs. Condition"))

q16_2 <- (ggplot(subdata16)
        + geom_point(aes(y=condition, x=price))
        + theme_bw()
        + labs(title = "Price vs. Condition"))

q16_3 <- (ggplot(subdata16)
        + geom_point(aes(y=condition, x=year))
        + theme_bw()
        + labs(title = "Year vs. Condition"))
```

Part 2:

```
##=========Anomalies=========

###### 1 - Odometer Readings

#Looking at very high odometer readings
odread <- vposts[ , c("year", "odometer")]
```

```
par(mfrow = c(2,2))
#Plot once with no limit on odometer reading
plot(odread, main = "Odometer Readings by Year", xlab = "Year", ylab = "Odometer",
    xlim = c(1920, 2016), pch = 20)
#Plot without the max odometer reading
max(odread, na.rm = TRUE)
plot(odread, main = "Odometer Readings by Year - 2", xlab = "Year", ylab = "Odometer",
    xlim = c(1920, 2016), ylim = c(0, 99999999), pch = 20)
#Plot without odometer readings above 500,000
odread2 <- odread[order(-odread$odometer),]
odread2[1:20,]
plot(odread, main = "Odometer Readings by Year - 3", xlab = "Year", ylab = "Odometer",
    xlim = c(1920, 2016), ylim = c(0, 500000), pch = 20)

#Looking at odometer readings of zero
odreadlow <- vposts[order(vposts$odometer), ]
nrow(vposts[which(vposts$odometer == 0),])
odreadzero <- odreadlow[c("city", "odometer")][1:89,]
byCity <- lapply(unique(odreadzero$city), function(x) odreadzero[which(odreadzero$city == x),
])
counts <- sapply(1:7, function(x) nrow(byCity[[x]]))
names(counts) <- unique(vposts$city)
counts

#Check to see if these were updated recently. Shows that they wanted to sell them quickly.
odreadzeropost <- odreadlow[c("city", "posted", "updated")][1:89,]
odreadzerobody <- odreadlow[c("body")][1,]

###### 2 - Posted vs. Updated Time
wrongtime <- vposts[which(vposts$updated < vposts$time), c("time", "updated", "city")]
nrow(wrongtime)
head(wrongtime)
timeByCity <- lapply(unique(wrongtime$city), function(x) wrongtime[which(wrongtime$city
== x), ])
counts <- sapply(1:7, function(x) nrow(timeByCity[[x]]))
names(counts) <- unique(vposts$city)
counts

###### 3 - Year
minyear <- vposts[which(vposts$year == 4), ]
maxyear <- vposts[which(vposts$year > 2016), ]
cond <- 4 < vposts$year & vposts$year <= 2016
no_outliers <- vposts[cond, ]
min(vposts$year)
min(no_outliers$year)
```

```
secondmin <- no_outliers[which(no_outliers$year == 1900),]
secondmin[,c("city", "price", "year", "title")][1:nrow(secondmin), ]
cond2 <- 1900 < vposts$year & vposts$year <= 2016
no_outliers2 <- vposts[cond2, ]
min(no_outliers2$year)

#Plot 3 histograms with less years in each
par(mfrow = c(2,2))
hist(no_outliers2$year, main = "Distribution of Years: 1921-2016",
    xlab = "Years", ylab = "Number of Vehicles", xlim = c(1921,2016), breaks = 95)

smallrange <- no_outliers[which(no_outliers$year >= 1960),]
hist(no_outliers2$year, main = "Distribution of Years: 1960-2016",
    xlab = "Years", ylab = "Number of Vehicles", xlim = c(1960,2016), breaks = 56)

smallerrange <- no_outliers[which(no_outliers$year >= 1990),]
hist(smallerrange$year, main = "Distribution of Years: 1990-2016",
    xlab = "Years", ylab = "Number of Vehicles", xlim = c(1990,2016), breaks = 26)

##==========Insights==========

library(ggplot2)

subdata <- vposts[vposts$price < 500000, ]
subdata <- subdata[subdata$year > 1920, ]

###### 1 - Price by Year in Each City
#Using geom_smooth which automatically uses a general additive model
#for a dataset of this size.
PbyCity <- (ggplot(subdata)
            + geom_smooth(aes(x = year, y = price, color = city), se = FALSE)
            + labs(title = "Price by Year in Each City", x = "Year", y = "Price"))
PbyCity

###### 2 - Average Odometer by Type and by Year
subdata2 <- vposts[vposts$odometer< 500000, ]
subdata2 <- subdata2[subdata2$type != "", ]

avgOdbyType <- (ggplot(subdata2)
             + geom_boxplot(aes(x = type, y = odometer))
             + labs(title = "Average Odometer by Type", x = "Type", y = "Odometer"))
avgOdbyType

#Look at average by year
subdataclean <- subdata2[subdata2$year <= 2015, ]
subdataclean <- subdataclean[subdataclean$year > 1920, ]
```

```
byYear <- (ggplot(subdataclean)
          + geom_boxplot(aes(x = year, y = odometer, fill = factor(year)))
          + labs(title = "Average Odometer by Year", x = "Year", y = "Odometer"))
byYear

###### 3 - Odometer vs. Fuel Type
#omitting all the rows with NA in them also omitted all the electric cars.
#but this is expected since all electric cars do not have cylinders.
subdata3 <- subdata2[subdata2$fuel != "", ]
odvfuel <- (ggplot(subdata3)
           + geom_boxplot(aes(x = fuel, y = odometer))
           + labs(title = "Average Odometer by Fuel", x = "Fuel", y = "Odometer"))
odvfuel
```

Part 3:

```
library(stringr)
body = vposts$body

# ===== Q1 =====
# Extract prices

prices = vposts$price
prices = sapply(prices, function(x){
  if(!is.na(x)){
    as.numeric(x)
  }
  else{
    return(0)
  }
})

# Find the prices in the body column
re = '(\\$)([0-9\\,]+)'
matches = str_extract(body, re)
matches = gsub("\\$", "", matches)
matches = gsub("\\,", "", matches)
matches = sapply(matches, function(x){
  if(!is.na(x)){
    as.numeric(x)
  }
  else{
    return(0)
  }
})
matches = as.numeric(matches)
```

```
# Find number that match the price column
agree = as.numeric(unlist(mapply(agrep, matches, prices, value = TRUE)))
length(agree)/length(prices)

# ===== Q2 =====
# Extract VIN numbers from the body
re2 = 'VIN\\: ([0-9a-zA-Z]{17})'
vin = str_extract(body, re2)
vin = gsub("VIN\\: ", "", vin)
numposts = sapply(vin, function(x){
  if(!is.na(x)){
    return(x)
  }
})
numposts = unlist(numposts)
numposts[1:100] #check that vin numbers are good
length(numposts) #get number of posts
vposts.2 = cbind(vposts, vin)

# ===== Q3 =====
# Extract phone numbers from the body
re3 = '(\\()?[0-9]{3}[:punct:]*[:space:]?[0-9]{3}(-)*[0-9]{4}'
phone = str_extract(body, re3)
phone[2000:22000]
numposts2 = sapply(phone, function(x){
  if(!is.na(x)){
    return(x)
  }
})
numposts2 = unlist(numposts2)
numposts2[1:300] #check that all phone numbers are apporpriate
length(numposts2) # get number of posts
vposts.2 = cbind(vposts.2, phone)

# ===== Q4 =====
# Extract emails from the body
re4 = '([a-zA-Z0-9_.-]+)@([a-zA-Z0-9]+).(com|COM|edu|EDU|net|NET|org|ORG)'
email = str_extract(body, re4)
numposts3 = sapply(email, function(x){
  if(!is.na(x)){
    return(x)
  }
})
numposts3 = unlist(numposts3)
numposts3 #check that all the emails are correct
length(numposts3)
```

```
vposts.2 = cbind(vposts.2, email)

# ===== Q5 =====
# Extract year from the body
re5 = '[1|2][0|9][0-9]{2}'
year1 = str_extract(body, re5)
year1 = as.integer(year1)
numposts4 = sapply(year1, function(x){
  if(!is.na(x)){
    return(1)
  }
  else{
    return(0)
  }
})
table(numposts4)

# Find number that match the year column
year = vposts$year
agree2 = as.numeric(unlist(mapply(agrep, year1, year, value = TRUE)))
length(agree2)/length(year)

# Extract year from the description column
description = vposts$description
re5.1 = '^[1|2][0|9][0-9]{2}'
year2 = str_extract(description, re5.1)
year2 = as.integer(year2)
numposts4.1 = sapply(year2, function(x){
  if(!is.na(x)){
    return(1)
  }
  else{
    return(0)
  }
})
table(numposts4.1)
agree2.1 = as.numeric(unlist(mapply(agrep, year2, year, value = TRUE)))
length(agree2.1)/length(year)

# ===== Q6 =====
# Extract the model of the car
makers = unique(vposts$maker)
title = vposts$title
re6 = '^[0-9]{4} [a-zA-Z]+ [a-zA-Z0-9]+'
model = str_extract(title, re6)
model = gsub('^[0-9]{4} [a-zA-Z]+ ', '', model)
```

```
numposts5 = sapply(model, function(x){
  if(!is.na(x)){
    return(x)
  }
})
numposts5 = as.character(unlist(numposts5))
numposts5 #check that all the models are correct
length(numposts5)
head(numposts5)

# ===== Modeling ======
# +++++ Toyotas +++++
toyotas = vposts[which(vposts$maker == "toyota"),
                 c("odometer", "year", "condition", "city", "price")]
fit = lm(price ~ odometer + year + condition + city, data = toyotas)
summary(fit)
anova(fit)
plot(fit)
fit1 = lm(price ~ odometer + year + city, data = toyotas)
summary(fit1)
anova(fit1)
fit2 = lm(price ~ odometer + year + condition, data = toyotas)
summary(fit2)
anova(fit2)
anova(fit2, fit)

# +++++ BMW +++++
bmw = vposts[which(vposts$maker == "bmw"),
                 c("odometer", "year", "condition", "city", "price")]
fit = lm(price ~ odometer + year + condition + city, data = bmw)
summary(fit)
anova(fit)
fit1 = lm(price ~ odometer + year + city, data = bmw)
summary(fit1)
anova(fit1)
fit2 = lm(price ~ odometer + year + condition, data = bmw)
summary(fit2)
anova(fit2)
anova(fit2, fit)
```