STA141 Assignment 1 I

Chad Pickering

Monday, October 05, 2015

Chad Pickering A03 | 913328497 | 10/5/2015

Corresponded with: Janice Luong, Rico Lin, Ricky Safran, Sierra Tevlin, Hannah Kosinovsky

Resources Used: Piazza forums, office hours, statmethods.net, R Programming text (Roger Peng)

```
print(load("C:/Users/cpickering/Syncplicity Folders/ChadSync/STATISTICS/STA141/vehicles.rda"))
```

```
## [1] "vposts"
```

1.

```
nrow(vposts)
```

```
## [1] 34677
```

The number of observations in the dataset is 34677.

2.

```
names(vposts) #names of columns
```

```
##  [1] "id"           "title"        "body"         "lat"
##  [5] "long"         "posted"       "updated"      "drive"
##  [9] "odometer"     "type"         "header"       "condition"
## [13] "cylinders"    "fuel"         "size"         "transmission"
## [17] "byOwner"      "city"         "time"         "description"
## [21] "location"     "url"          "price"        "year"
## [25] "maker"        "makerMethod"
```

```
lapply(vposts, class) #loop over all vposts column names, give class
```

```
## $id
## [1] "character"
##
## $title
## [1] "character"
##
## $body
## [1] "character"
```

```
##
## $lat
## [1] "numeric"
##
## $long
## [1] "numeric"
##
## $posted
## [1] "POSIXct" "POSIXt"
##
## $updated
## [1] "POSIXct" "POSIXt"
##
## $drive
## [1] "factor"
##
## $odometer
## [1] "integer"
##
## $type
## [1] "factor"
##
## $header
## [1] "character"
##
## $condition
## [1] "factor"
##
## $cylinders
## [1] "integer"
##
## $fuel
## [1] "factor"
##
## $size
## [1] "factor"
##
## $transmission
## [1] "factor"
##
## $byOwner
## [1] "logical"
##
## $city
## [1] "factor"
##
## $time
## [1] "POSIXct" "POSIXt"
##
## $description
## [1] "character"
##
## $location
## [1] "character"
```

```
##
## $url
## [1] "character"
##
## $price
## [1] "integer"
##
## $year
## [1] "integer"
##
## $maker
## [1] "character"
##
## $makerMethod
## [1] "numeric"
```

3.

```
mean(vposts$price, na.rm=TRUE) #all data; outliers included
```

```
## [1] 49449.9
```

```
median(vposts$price, na.rm=TRUE)
```

```
## [1] 6700
```

```
quantile(vposts$price, probs=seq(0,1,0.1), na.rm=TRUE) #deciles
```

```
##        0%       10%       20%       30%       40%       50%       60%
##         1      1200      2499      3500      4995      6700      8900
##       70%       80%       90%      100%
##     11888     15490     21997 600030000
```

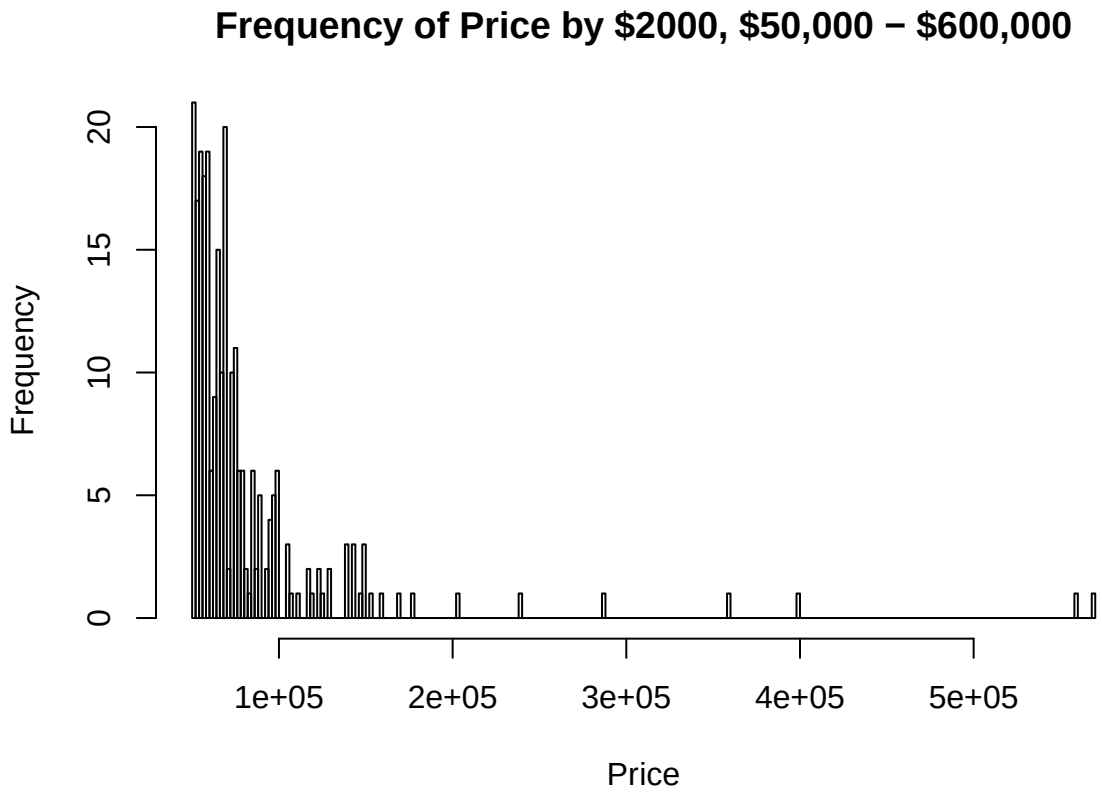
```
price_mean <- mean(vposts$price, na.rm=TRUE)
price_sd <- sd(vposts$price, na.rm=TRUE)
price_median <- median(vposts$price)
```

The original standard deviation is too large to use with Chebyshev's Theorem or the Empirical Rule, as the distribution is far too skewed; unfortunately I have to be slightly arbitrary.

```
par(mfrow=c(1,1), mar=c(5,5,3,2.1))
head(sort(vposts$price, decreasing=TRUE), 20) #true outliers appear to be 4 top prices, will exclude
```

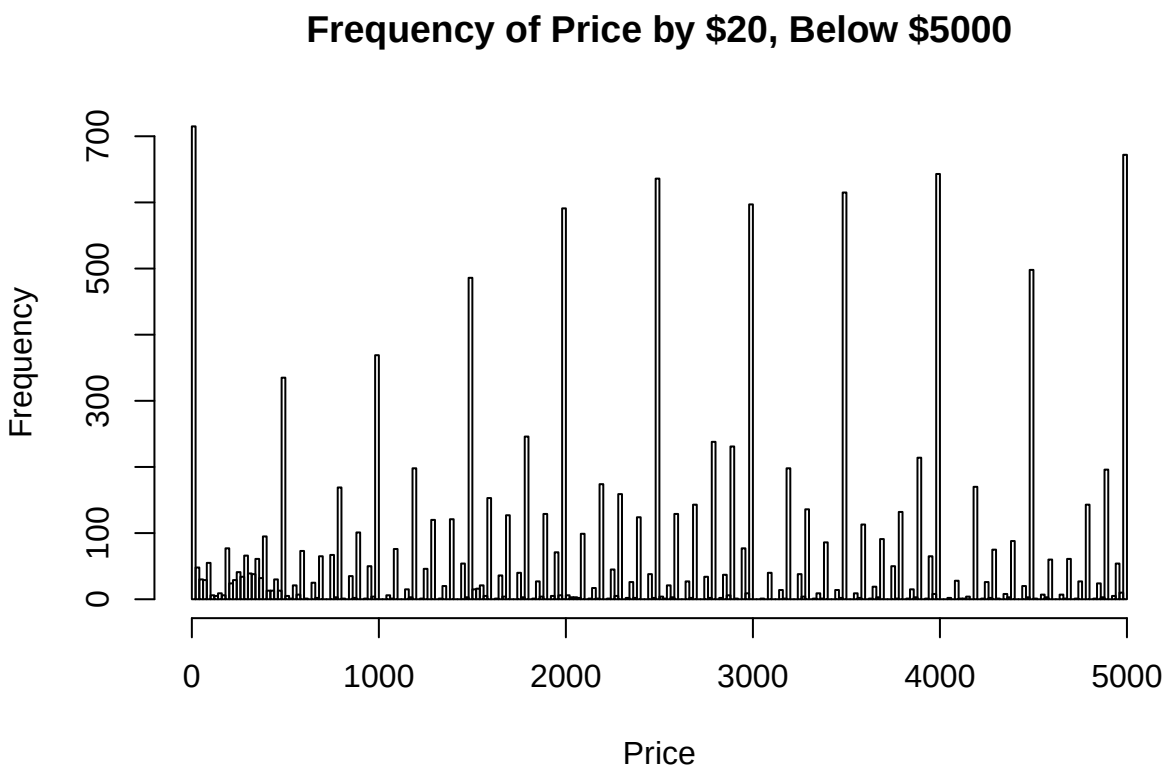
```
##  [1] 600030000 600030000  30002500   9999999    569500    559500    400000
##  [8]    359000    286763    240000    202455    177588    169000    159000
## [15]    152900    150000    149995    149890    147000    143950
```

```
highest_prices <- subset(vposts$price, vposts$price >= 50000 & vposts$price <= 600000)
hist(highest_prices, breaks=275, xlab="Price", main="Frequency of Price by $2000, $50,000 - $600,000")
```



It looks as though $600000 is a reasonable maximum - I cannot exclude prices that follow a reasonable spread in the right tail.

```
lowest_prices <- subset(vposts$price, vposts$price <= 5000)
hist(lowest_prices, breaks=250, xlab="Price", main="Frequency of Price by $20, Below $5000")
```



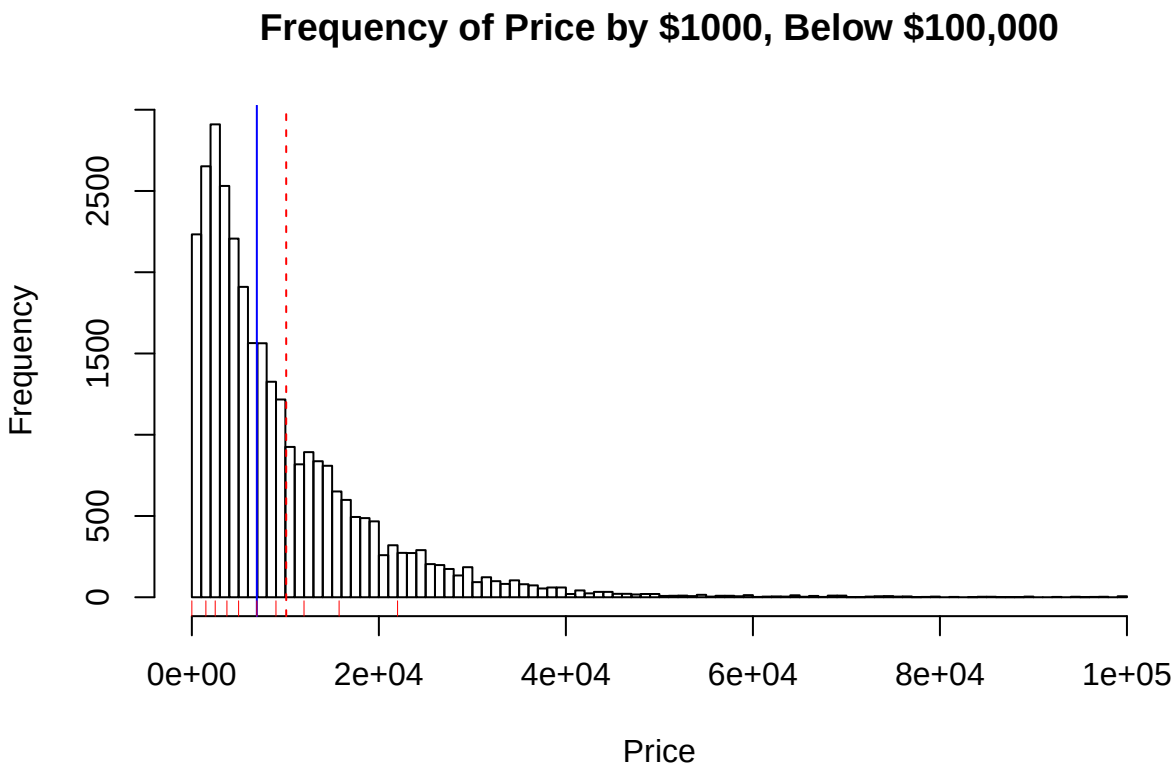
It looks as though I can safely exclude the $1 category; anything more are chiefly part(s) being sold by the owner.

The following are the mean, median, and deciles using the corrected subsetted prices, all significantly more reasonable to the scope and source of the data:

```
corrected_prices <- subset(vposts$price, vposts$price >= 2 & vposts$price <= 600000)
corrected_meanprice <- mean(corrected_prices, na.rm=TRUE)
corrected_medianprice <- median(corrected_prices, na.rm=TRUE)
corrected_pricedeciles <- quantile(corrected_prices, probs=seq(0,1,0.1), na.rm=TRUE) #deciles
```

See #8 for more about my decisions to replace the individual large outliers.

```
hist(corrected_prices[corrected_prices <= 100000],
     breaks=100, xlab="Price", main="Frequency of Price by $1000, Below $100,000") #corrected prices
abline(v=corrected_meanprice, col="red", lty="dashed") #mean is a vertical red dashed line
abline(v=corrected_medianprice, col="blue") #median is a vertical blue solid line
rug(corrected_pricedeciles, col="red", quiet=TRUE) #deciles are solid red tick marks along the x-axis
```



The histogram shows the mean greater than the median, a feature of skewed right distributions. A majority of cars are sold for between a few thousand to around $20000, while only more rare or newer cars are sold for more.

4.

The levels() function gives all categories present in the indicated column.

```
levels(vposts$type)
```

```
##  [1] "bus"         "convertible" "coupe"       "hatchback"   "mini-van"
##  [6] "offroad"     "other"       "pickup"      "sedan"       "SUV"
## [11] "truck"       "van"         "wagon"
```

If we were to include NA values, the type proportions would not add to 1 and would stand as follows:

```
type_na <- sum(is.na(vposts$type)==TRUE)
vehtype <- sort(table(vposts$type), decreasing=TRUE)
vehtype_prop <- vehtype/(nrow(vposts)-type_na) #excludes NAs from denominator, what prop.table does
vehtype_prop_all <- vehtype/(nrow(vposts)) #includes NAs in denominator
round(vehtype_prop_all,4)
```

```
##
##       sedan         SUV       coupe       truck      pickup   hatchback
```

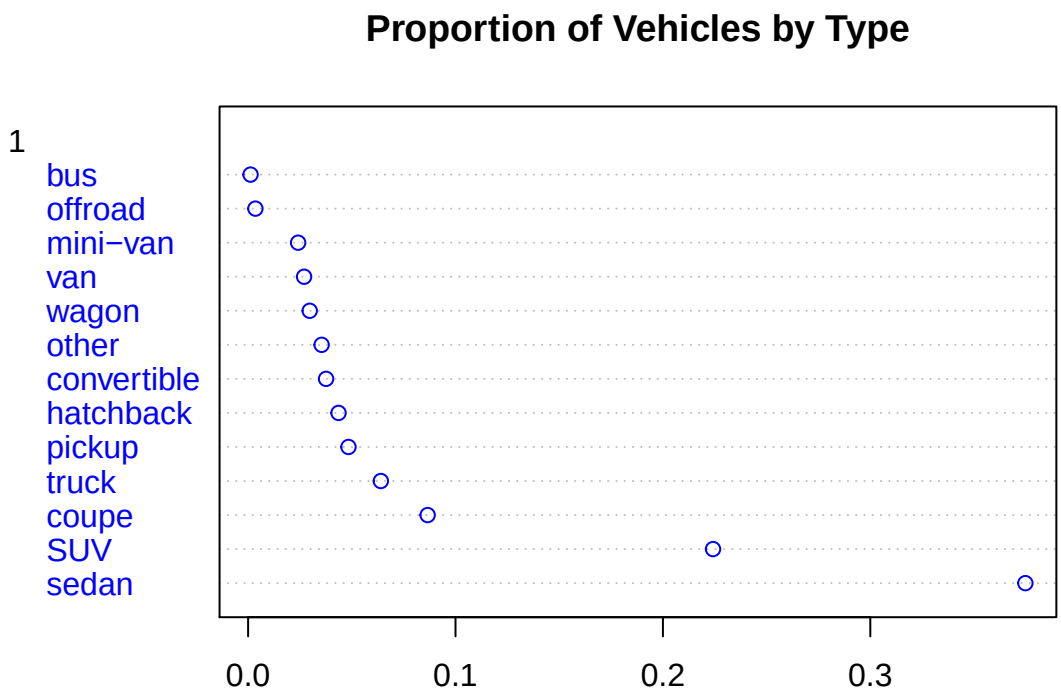
```
##      0.2030      0.1214      0.0469      0.0347      0.0262      0.0236
## convertible       other       wagon         van    mini-van     offroad
##      0.0204      0.0192      0.0161      0.0146      0.0131      0.0019
##         bus
##      0.0006
```

In this case, excluding the NA values yields more truthful proportions relative to the data, as they add to 1. We can then analyze the proportions of types that were truly reported:

```
trueprop <- sort(prop.table(table(vposts$type)), decreasing=TRUE)
round(trueprop,4)
```

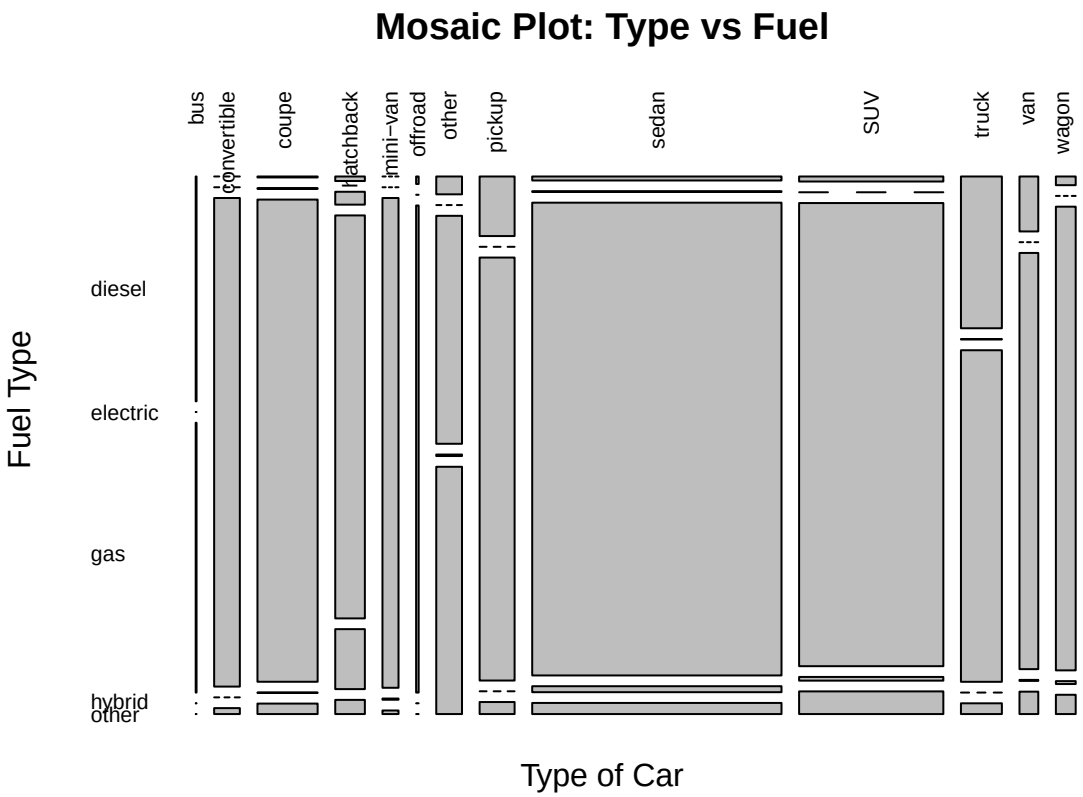
```
##
##       sedan         SUV       coupe       truck      pickup   hatchback
##      0.3748      0.2242      0.0866      0.0640      0.0484      0.0436
## convertible       other       wagon         van    mini-van     offroad
##      0.0376      0.0355      0.0297      0.0270      0.0241      0.0035
##         bus
##      0.0012
```

```
dotchart((as.matrix(trueprop)), col="blue", lwd=15,  main="Proportion of Vehicles by Type")
```

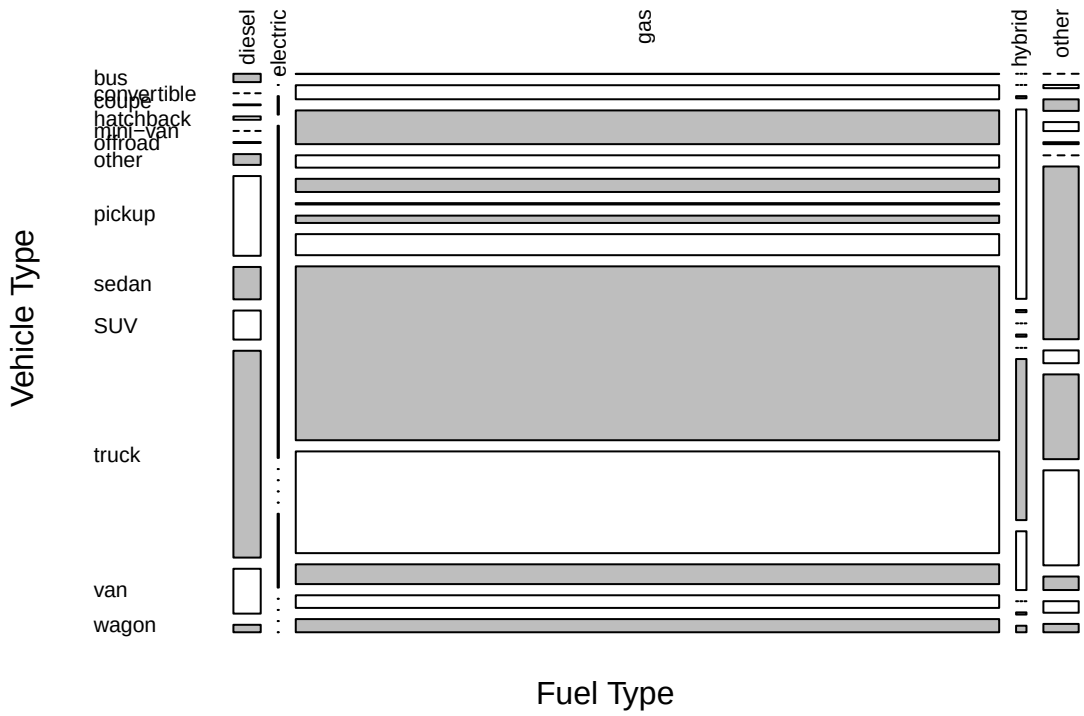


5.

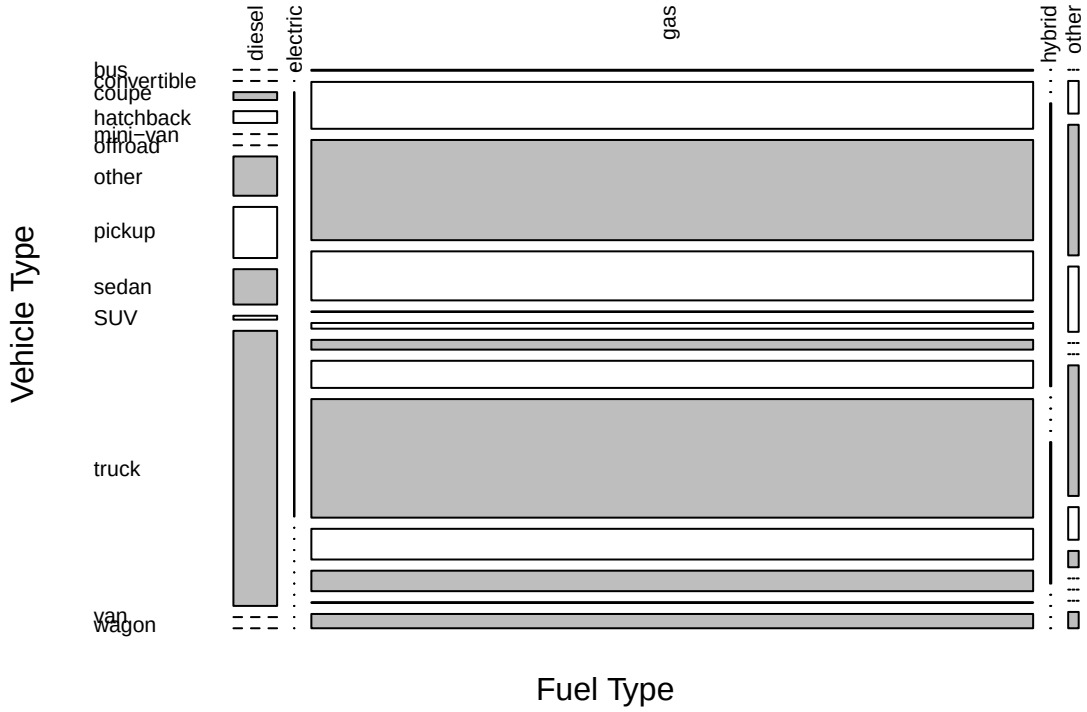
```
fueltypetable <- table(vposts$type, vposts$fuel)
par(mfrow=c(1,1), mar=c(3,3,3,2.1))
mosaicplot(fueltypetable, xlab="Type of Car", ylab="Fuel Type",
           main="Mosaic Plot: Type vs Fuel", las=2)
```



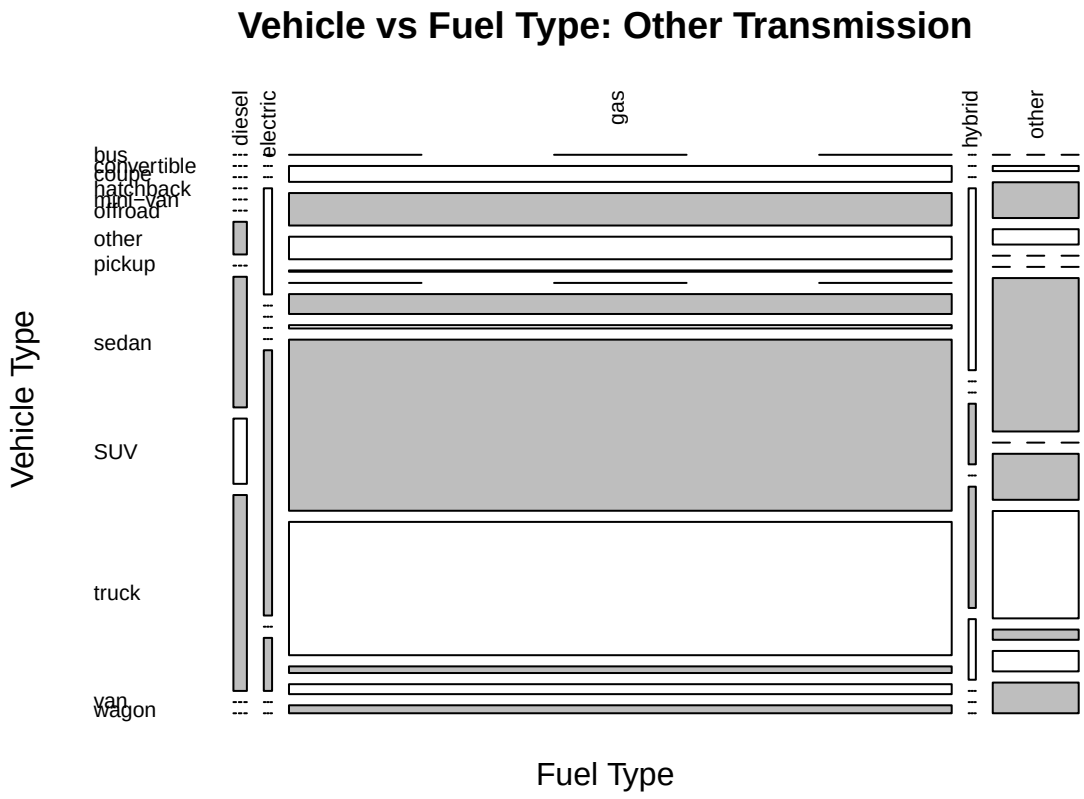

```
transmission_5 <- split(vposts, vposts$transmission)
table_5 <- lapply(transmission_5, function(d) table(d$fuel, d$type))

mosaicplot(table_5$automatic, col = c("gray", "white"),
           xlab="Fuel Type", ylab="Vehicle Type",
           main="Vehicle vs Fuel Type: Automatic Transmission", las=2)
```




```
mosaicplot(table_5$manual, col = c("gray", "white"),
           xlab="Fuel Type", ylab="Vehicle Type",
           main="Vehicle vs Fuel Type: Manual Transmission",las=2)
```




```
mosaicplot(table_5$other, col = c("gray", "white"),
           xlab="Fuel Type", ylab="Vehicle Type",
           main="Vehicle vs Fuel Type: Other Transmission",las=2)
```



The mosaic plots show that automatic vehicles tend to be gas-powered, while diesel, electric, hybrid, and other categories are relatively small. Electric and hybrid fuel types are more common with automatic vehicles (tend to be more recent), with electric and hybrid vehicles only consuming an almost negligible fraction of the manual plot. Additionally, the distribution of vehicle types in the manual plot look more uniform than they do in the automatic plot, especially apparent in the gas category. The group "other" has a high rate of "other" fuel types, raising questions about the specifics there. Without using any standard hypothesis testing methods, I will make the assertion that transmission type does play a role in the variation; however, it would likely not be a primary factor. The mosaic plot generated by aggregating all columns mentioned is incomprehensible and is excluded.

6.

Because of the small scope of the question, I made vposts$city its own data.frame.

```
cityfreq <- as.data.frame(table(vposts$city))
nrow(cityfreq)
```

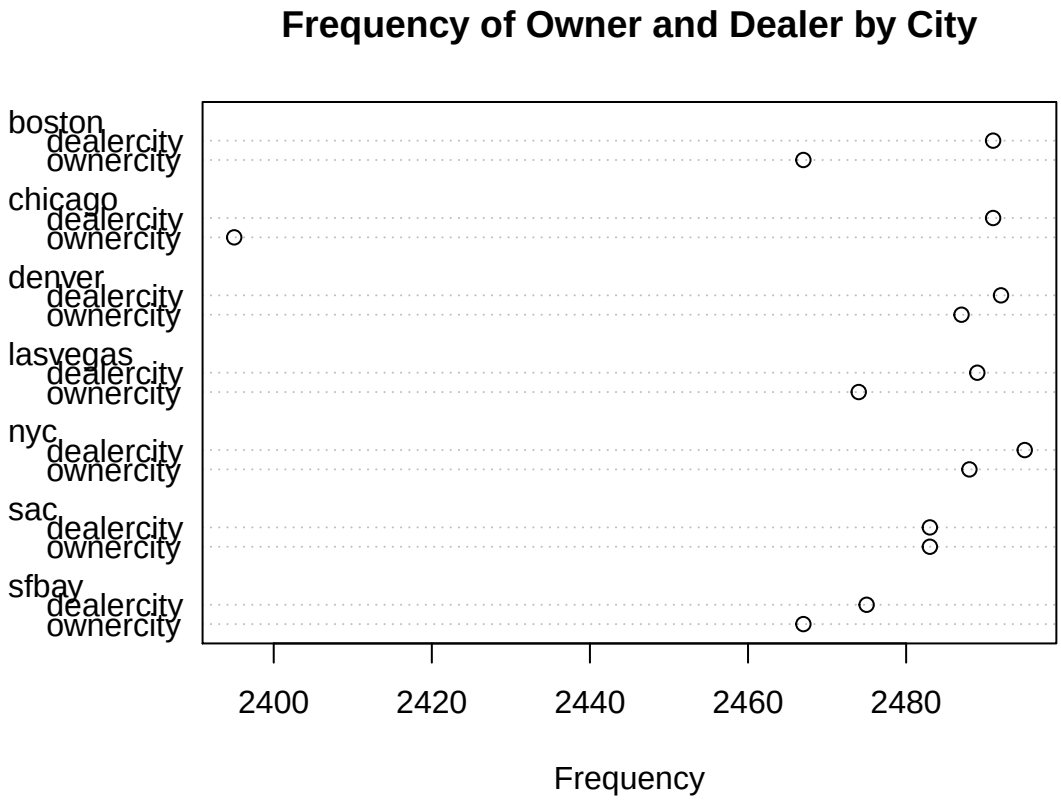
```
## [1] 7
```

7.

```r
byowner <- subset(vposts, byOwner==TRUE) #17261
bydealer <- subset(vposts, byOwner==FALSE) #17416
ownercity <- table(byowner$city) #subset byowner by city
dealercity <- table(bydealer$city) #subset byowner by city
compcity <- rbind(ownercity, dealercity)
```

If the data is examined with a dotchart and with the correct limits, the difference between the 'sold by' methods per city become significantly more clear.

```r
par(mfrow=c(1,1), mar=c(4.3,4.1,4.1,2.1))
dotchart(compcity, xlab="Frequency", main="Frequency of Owner and Dealer by City")
```



The even proportions suggest that all observations were chosen with a sales frequency limit in mind.

8.

```r
max(vposts$price, na.rm=TRUE) #gives 600030000
```

```
## [1] 600030000
```

```r
head(sort(vposts$price, decreasing=TRUE),4) #identifies the 4 outliers with contrast to next values;
```

```
## [1] 600030000 600030000  30002500   9999999
```

```r
                              # with original data, is 600030000, 600030000, 30002500, 9999999
which(vposts$price == max(vposts$price, na.rm=TRUE)) #rows of max values,
```

```
## [1] 4741 4880
```

```r
                              #use in an iterative fashion to correct outliers
vposts[4741, "price"] <- 30000 #set first of two identical outliers to higher of entry attempts
vposts[4880, "price"] <- 6000 #set second of two identical outliers to lower of entry attempts
vposts[8140, "price"] <- 2750 #set third large outlier to average of entry attempts
vposts[13937, "price"] <- NA #set fourth large outlier to NA
price_nooutliers <- subset(vposts$price, vposts$price >= 2 & vposts$price <= 600000)
                              #subset of data with no outliers, see #3 reasoning
max(price_nooutliers, na.rm=TRUE) #new maximum price (reasonable) - now 569500
```

```
## [1] 569500
```

The two identical outliers were posted a week apart, the second an update on price, marred by an entry error. Since the third large outlier was a single value, taking the average was best - it does not affect the distribution of prices to a noticeable level. The fourth large outlier was a string of seven 9s, which may be a default NA in the original database.

9.

```r
byowner <- subset(vposts, vposts$byOwner==TRUE) #17261; all byOwner
bydealer <- subset(vposts, vposts$byOwner==FALSE) #17416; all byDealer
byowner_makercity <- table(byowner$maker, byowner$city) #maker (y), city (x) [for byOwner]
bydealer_makercity <- table(bydealer$maker, bydealer$city) #maker (y), city (x) [for byDealer]
```

The three most common makes of cars for each city that are sold by the owner:

```r
top3_byowner <- function(count_city) {
  top3_o <- order(count_city, decreasing = TRUE)[1:3]
  rownames(byowner_makercity)[top3_o]
}
apply(byowner_makercity, 2, top3_byowner)
```

```
##
##        boston      chicago     denver      lasvegas    nyc
##   [1,] "ford"      "chevrolet" "ford"      "ford"      "nissan"
##   [2,] "honda"     "ford"      "chevrolet" "chevrolet" "toyota"
##   [3,] "chevrolet" "honda"     "toyota"    "toyota"    "honda"
##
##        sac         sfbay
##   [1,] "toyota"    "toyota"
##   [2,] "ford"      "honda"
##   [3,] "chevrolet" "ford"
```

The three most common makes of cars for each city that are sold by a dealer:

```
top3_bydealer <- function(count_city) {
  top3_d <- order(count_city, decreasing = TRUE)[1:3]
  rownames(bydealer_makercity)[top3_d]
}
apply(bydealer_makercity, 2, top3_bydealer)
```

```
##
##        boston      chicago     denver      lasvegas    nyc
##   [1,] "ford"      "chevrolet" "ford"      "ford"      "nissan"
##   [2,] "toyota"    "ford"      "chevrolet" "nissan"    "toyota"
##   [3,] "chevrolet" "nissan"    "dodge"     "chevrolet" "honda"
##
##        sac         sfbay
##   [1,] "ford"      "toyota"
##   [2,] "toyota"    "ford"
##   [3,] "chevrolet" "bmw"
```

The results showed that for each respective city, the top 3 had at least 2 in common with each other, not necessarily in the same order. Only New York City had a perfect match between sales methods. I think that the sales method makes minimal difference, most of which was caused by randomness. What was not caused by randomness, however, was the fact that cars sold by the owner tended to be cars that were older (#10), and so the corresponding makes of those cars were more frequent.

10.

```
which(vposts$year == 2022)
```

```
## [1] 21975
```

```
vposts[21975, "year"] <- 2002 #adjusting the year 2022; keystrokes similar to 2002
which(vposts$year == 4)
```

```
## [1] 8417
```

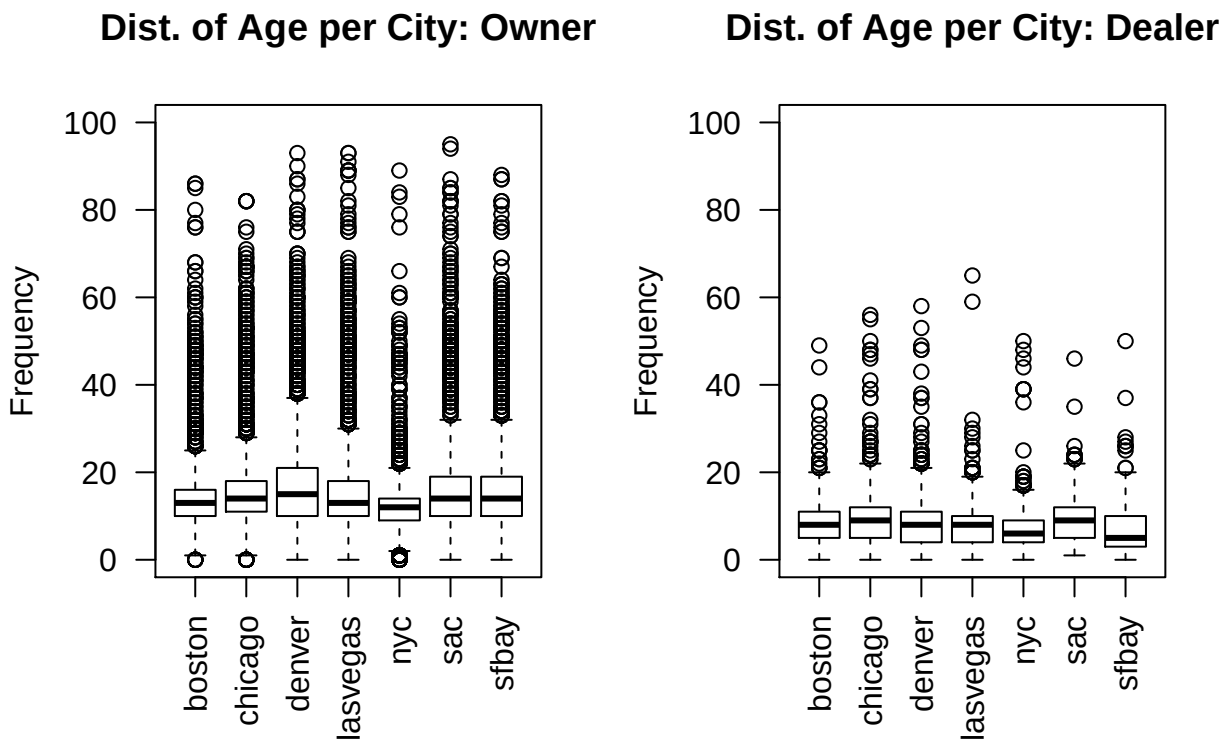
```
vposts[8417, "year"] <- 2004 #adjusting to the year written in the description
vposts$age <- 2016 - vposts$year
head(sort(vposts$age, decreasing=TRUE), 50) #7 vehicles from 1900 (sold in Sac) is rather curious
```

```
##  [1] 116 116 116 116 116 116 116  95  94  93  93  93  91  90  89  89  89
## [18]  89  89  88  88  87  87  87  87  87  86  86  86  85  85  85  85  84
## [35]  84  84  83  83  82  82  82  82  82  82  82  82  82  82  82  81
```

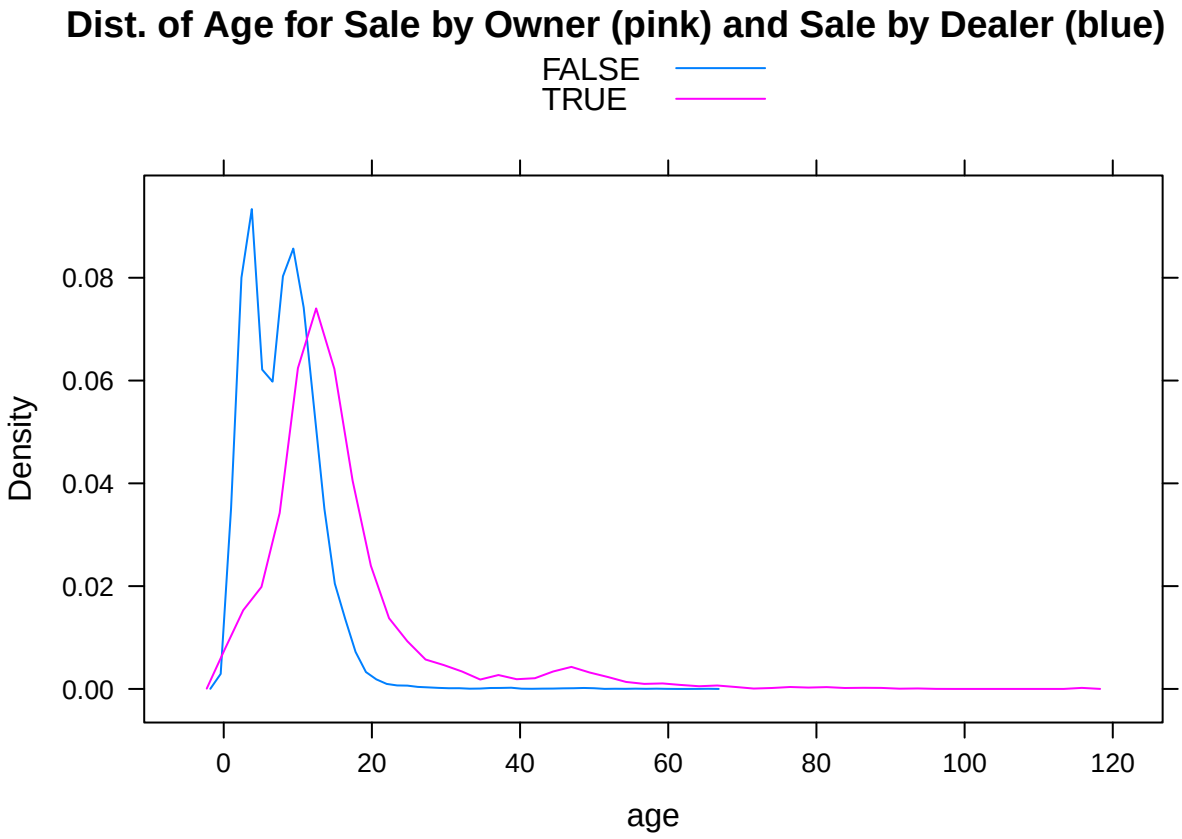
```
byowner_10 <- subset(vposts, byOwner==TRUE)
bydealer_10 <- subset(vposts, byOwner==FALSE)
par(mfrow = c(1,2), mar=c(6.1,4.1,4.1,2.1)) #for graphs, 1 row, 2 columns
plot(byowner_10$city, byowner_10$age,
```

```
     ylab="Frequency", ylim=c(0,100),
     main="Dist. of Age per City: Owner", las=2)
plot(bydealer_10$city, bydealer_10$age,
     ylab="Frequency", ylim=c(0,100),
     main="Dist. of Age per City: Dealer", las=2)

library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.1.3
```




```
par(mfrow=c(1,1), mar=c(5.1,4.1,4.1,2.1))
densityplot( ~ age , vposts, group = byOwner, plot.points=FALSE,
             main="Dist. of Age for Sale by Owner (pink) and Sale by Dealer (blue)", auto.key=TRUE)
```



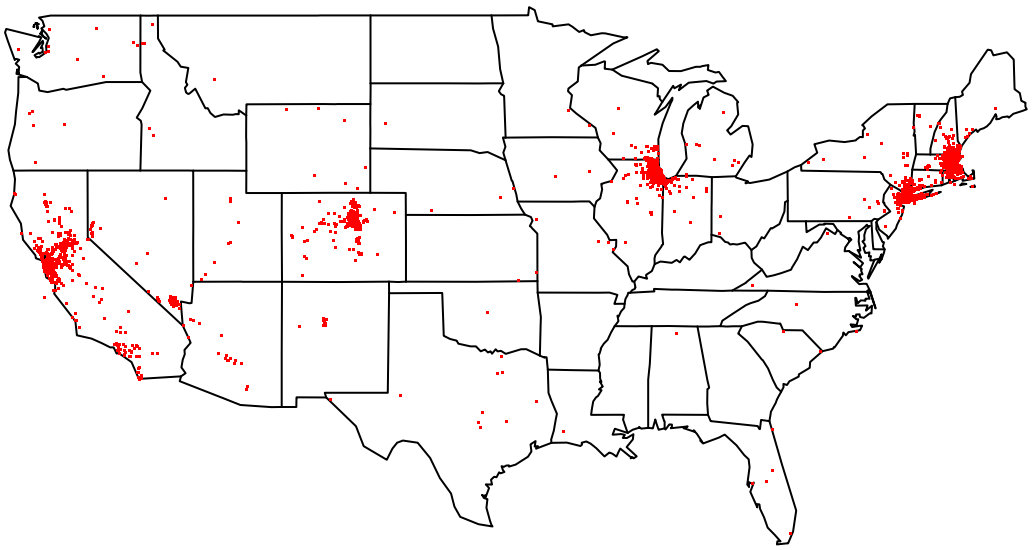
The side-by-side boxplots show one of the most stark and meaningful contrasts in this entire assignment. For all cities, vehicles sold by owners tend to be older in age than those sold by dealers. There are certainly more outlying values in the sale by owner plots, and higher medians for all cities in comparison with their 'by Dealer' counterparts. As seen in the density plot, sales by dealer seem to be bimodal and owner-originated deals are unimodal, skewed right, with a larger mean and median.

11.

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.1.3
```

```
map('state')
title="US Map: All Seller Locations"
points(vposts$long, vposts$lat, col="RED", pch=".")
```

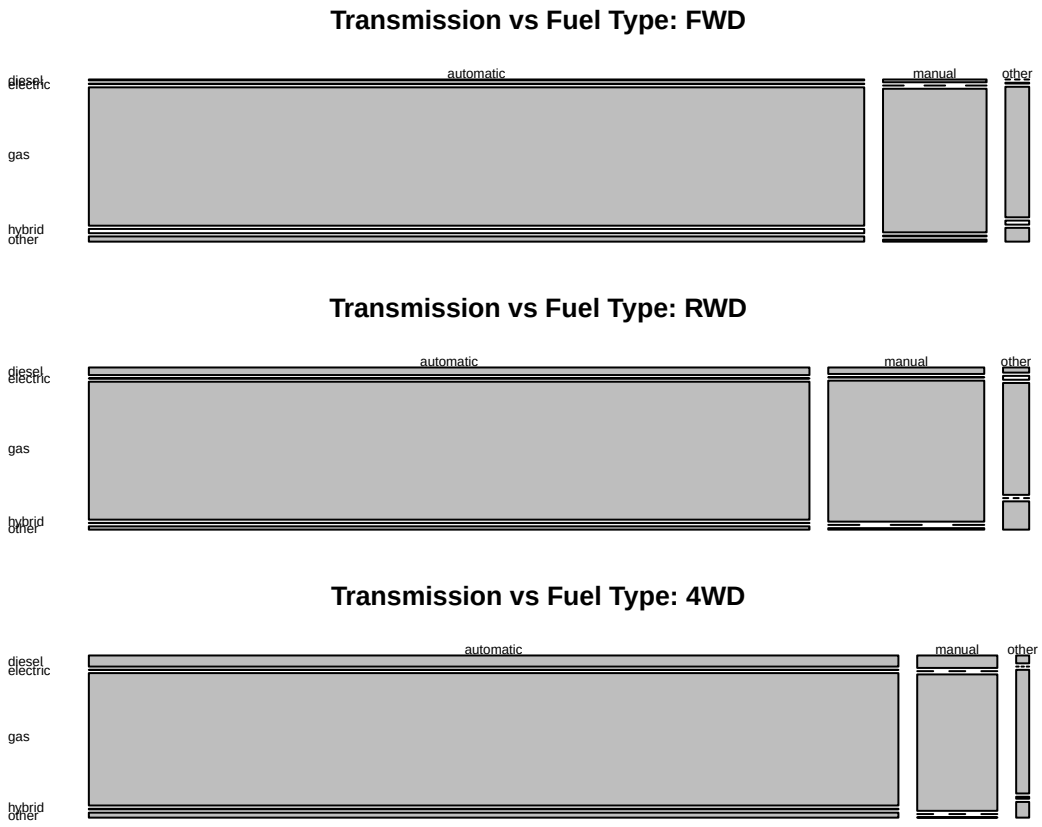


Locations of posts are concentrated in the more urban areas of the US, but more specifically around the 7 cities in the city column.

12.

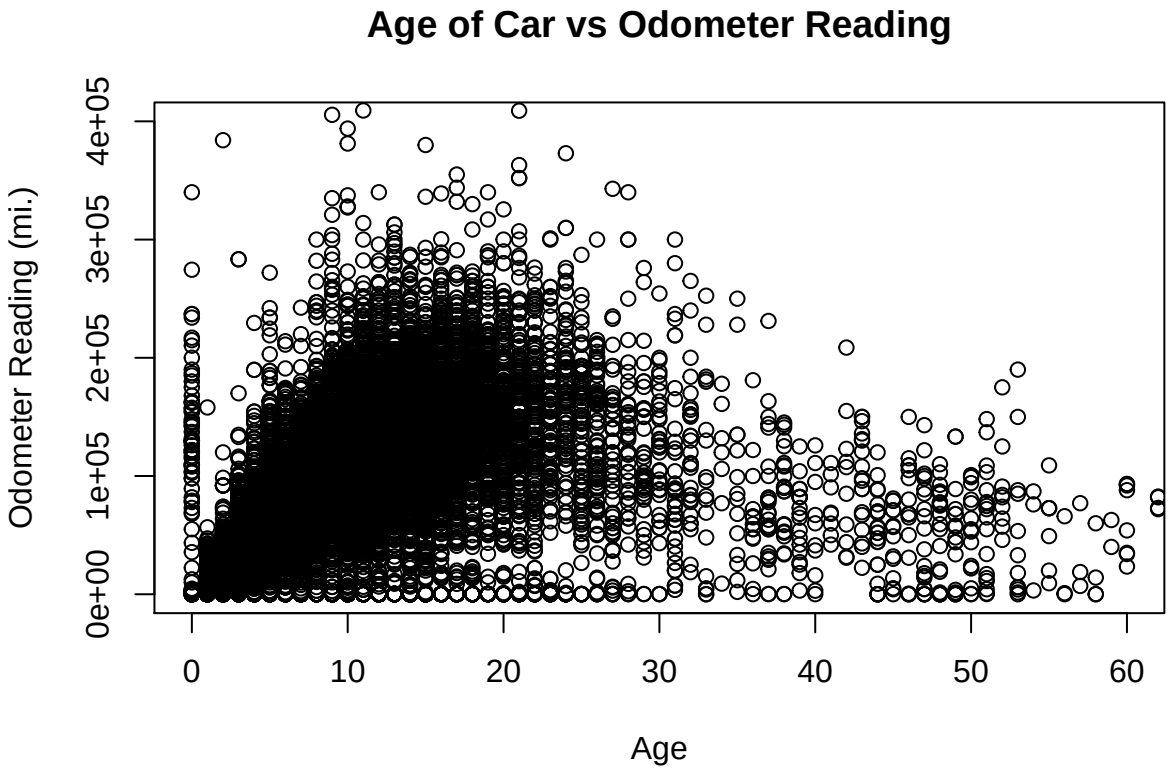
The mosaic plot generated by aggregating fuel, type, drive, and transmission is incomprehensible and is excluded.

```
drive_12 <- split(vposts, vposts$drive)
table_12 <- lapply(drive_12, function(d) table(d$transmission, d$fuel))

par(mfrow=c(3,1), mar=c(0.5,4.1,4.1,2.1))
mosaicplot(table_12$fwd, col = c("gray", "white"), main="Transmission vs Fuel Type: FWD",las=1)
mosaicplot(table_12$rwd, col = c("gray", "white"), main="Transmission vs Fuel Type: RWD",las=1)
mosaicplot(table_12$'4wd', col = c("gray", "white"),  main="Transmission vs Fuel Type: 4WD",las=1)
```



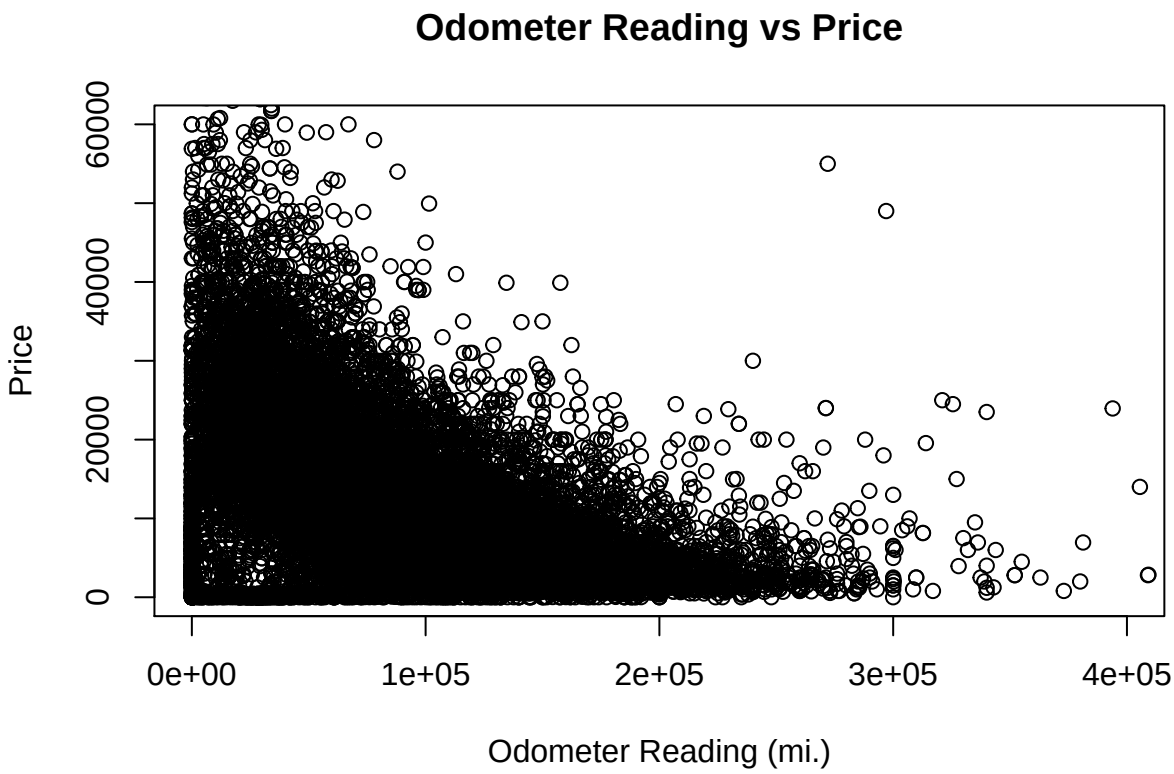
13.

```
vposts$age <- 2016 - vposts$year
par(mfrow=c(1,1), mar=c(5.1,4.1,4.1,2.1))
plot(vposts$age, vposts$odometer, xlab="Age", ylab="Odometer Reading (mi.)",
     main="Age of Car vs Odometer Reading", xlim=c(0,60), ylim=c(0, 400000)) #age vs odometer
```



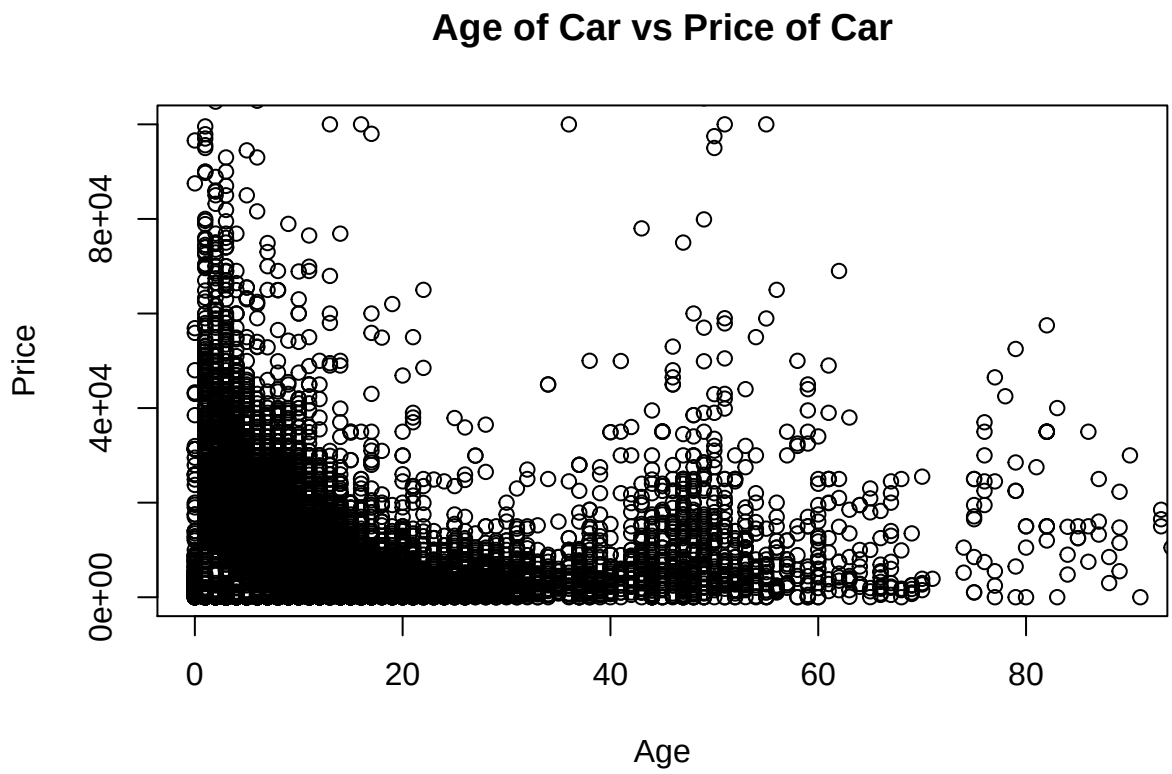
There is no linear correlation; however, more recent cars tend to have less miles on their odometers, but after around age 15, the median odometer reading appears to begin slowly receding, and cars that are greater than 40 years old tend to have no more than 100,000 miles.

```
plot(vposts$odometer, vposts$price, xlab="Odometer Reading (mi.)", ylab="Price",
     main="Odometer Reading vs Price", xlim=c(0,400000), ylim=c(0, 60000)) #price vs odometer
```



There is some correlation; density-wise, as price increases, odometer reading tends to decrease (range tends to decrease as well). Between the prices of about $1,000 and $10,000, we see a gap between those cars with ~0 miles and those with tens of thousands, similar to the prime gap.This gap disappears as the price increases above $10,000. This suggests that there are not many cars that are priced between ~$5,000 and $30,000 that have only a few thousand miles on them.

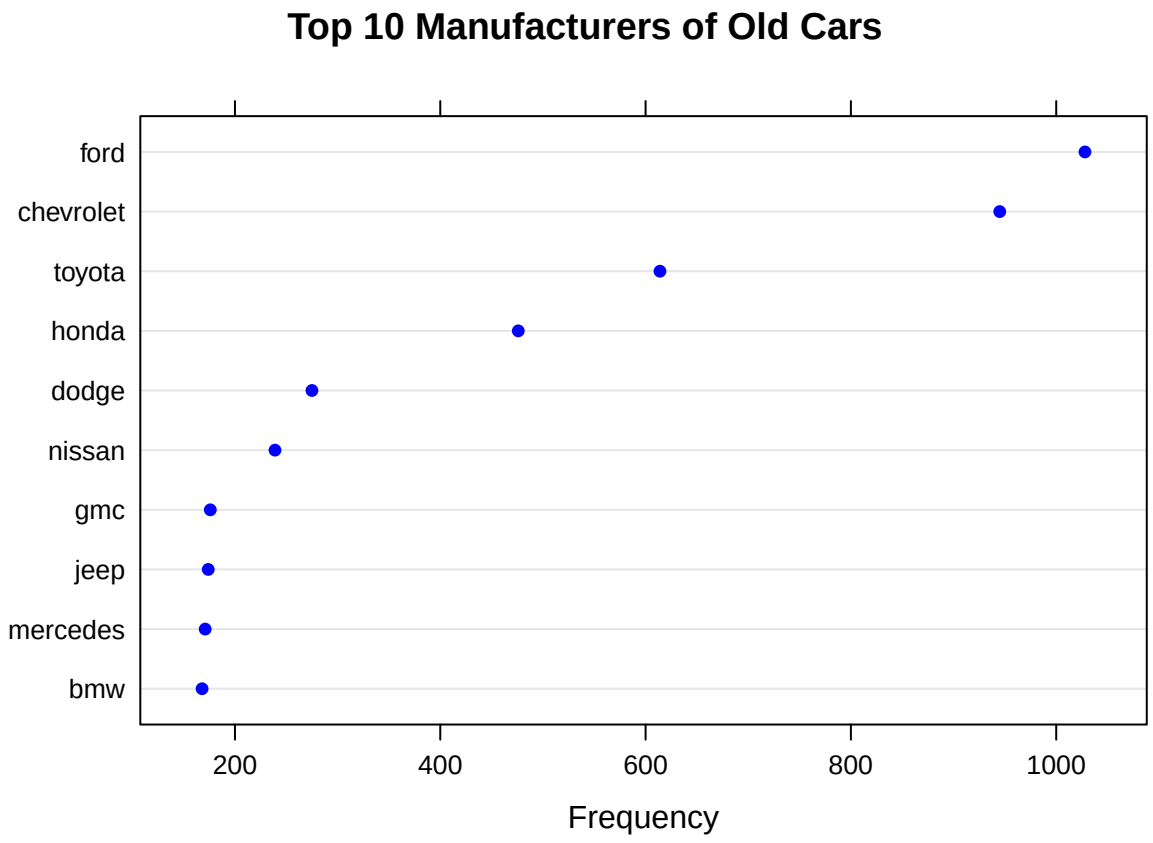
```
plot(vposts$age, vposts$price, xlab="Age", ylab="Price",
     main="Age of Car vs Price of Car", xlim=c(0,90), ylim=c(0, 100000)) #age vs price
```



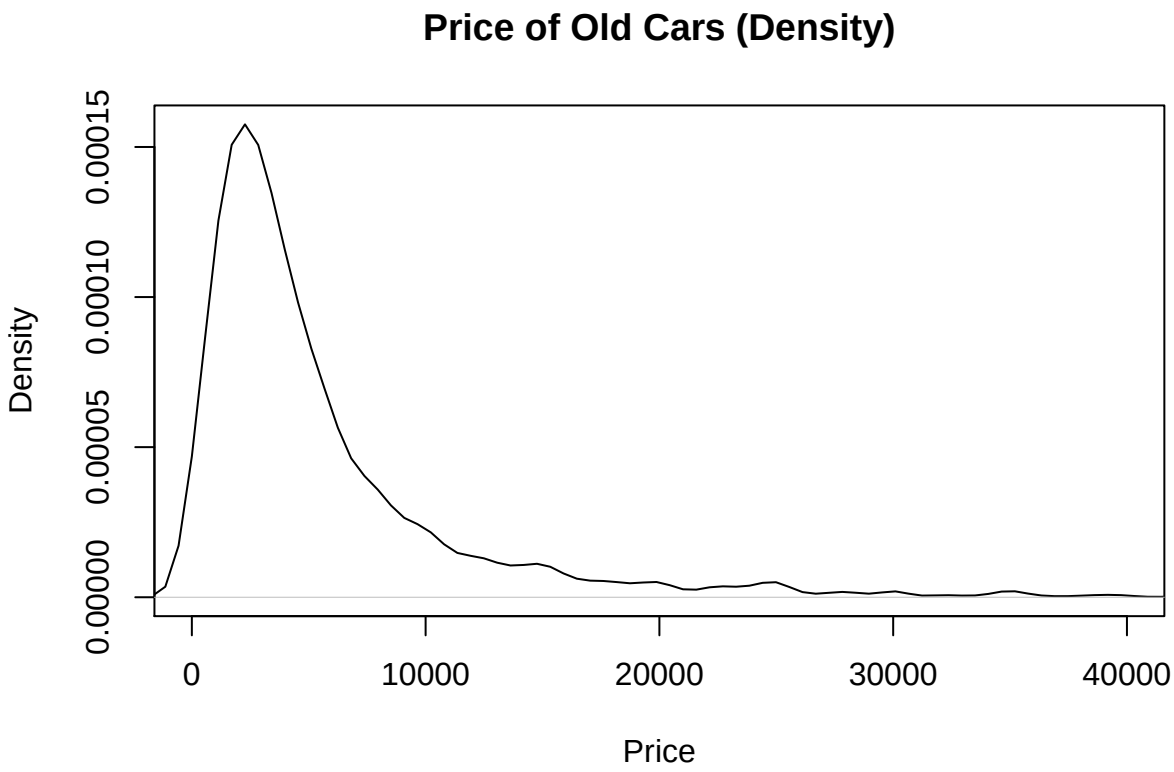
In general, as age increases, price exponentially decreases (until about 40 years). In the range of approximately 40 to 60 years, there is a small node of more expensive vehicles, representing classic/restored options from the 60s and 70s.

14.

```
par(mfrow=c(1,1), mar=c(5.1,4.1,4.1,2.1))
oldcars <- vposts[(vposts$year < 1995) | (vposts$odometer >= 150000), ] #definition of old car

oldmakerstop10 <- head(sort(table(oldcars$maker), decreasing=TRUE),10)
dotplot(sort(oldmakerstop10), col="blue", xlab="Frequency", main="Top 10 Manufacturers of Old Cars")
```




```
plot(density(oldcars$price, na.rm=TRUE),
     xlim=c(0,40000), xlab="Price", main="Price of Old Cars (Density)") #continuous dist.
```



Ford, Chevrolet, Toyota, and Honda manufactured the most of these cars, in that order. In the density plot, the distribution is skewed right, as the superset of the data is as well. Most "old cars" are priced less than $10,000, but some anomalies exist even past $30,000.

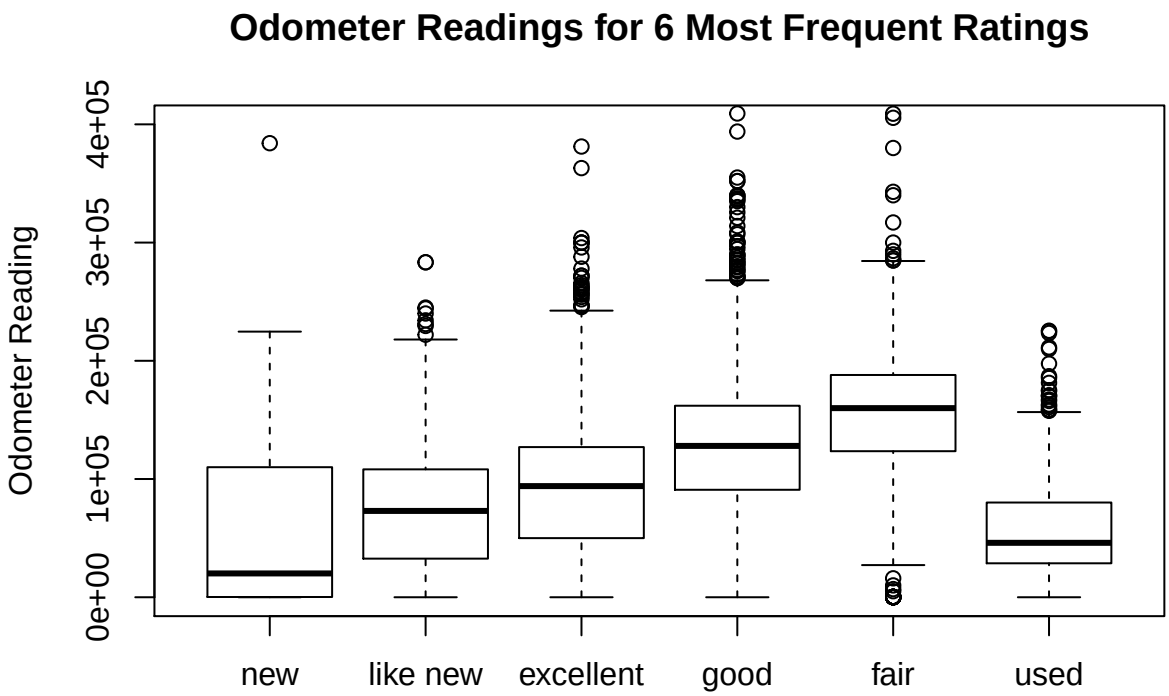
15.

Upon studying the raw dataset, a variable worthy of a separate column, preferably directly to the right of the "maker" column, is a "model" column. "Header" includes the year, maker, and model of the vehicle, but only year and maker have their own columns. In order to attain this data, we could parse the "header" column by looking for the row's corresponding maker in "header" and select everything in the cell after that. We could then cbind() our vector of results to the vposts dataset, or use my method from question 10. This could introduce the need for a miles per gallon variable for each year/maker/model triple.

16.

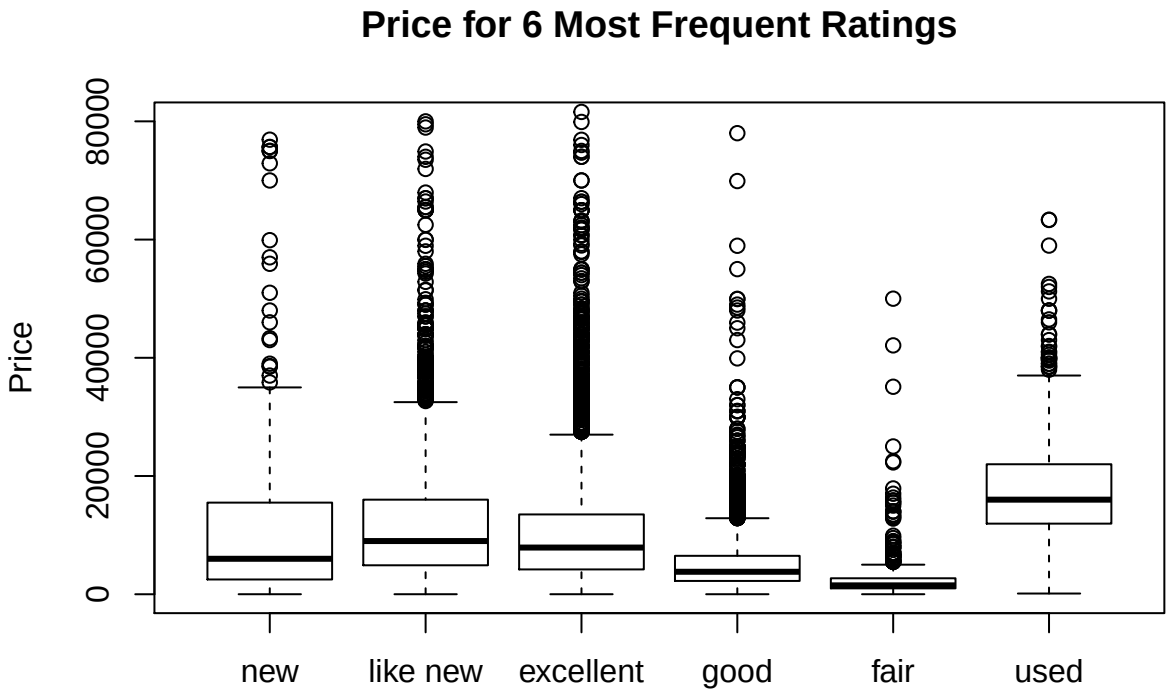
I am defining condition as only the 6 most frequent levels, in that exact order.

```
par(mfrow=c(1,1), mar=c(5.1,4.1,4.1,2.1))
odometer_cond <- split(vposts$odometer, vposts$condition)
boxplot(odometer_cond[c(22,13,7,10,8,42)], ylim=c(0,400000),
        ylab="Odometer Reading", main="Odometer Readings for 6 Most Frequent Ratings")
```



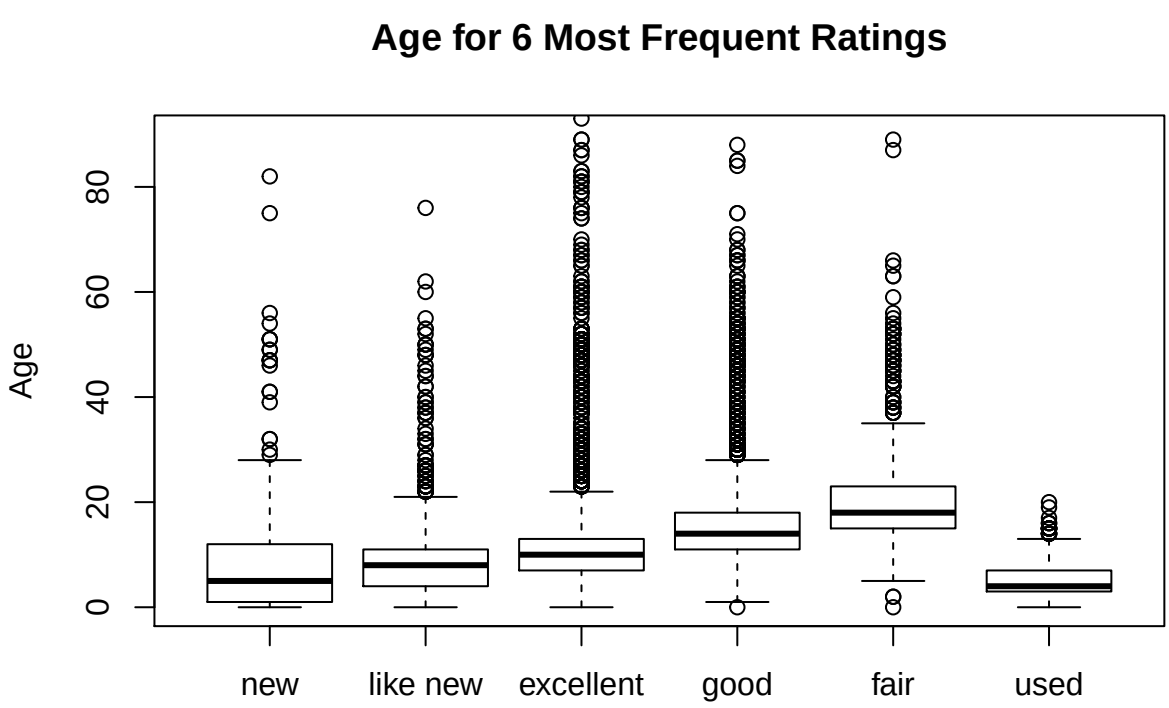
Among the 5 ratings that are not "used", it is shown that as the car acrues more miles, the rating is more likely to be worse. The medians and IQR ranges from "new" to "fair" increase for each level. What is interesting is that cars defined as "used" seem to have fewer miles on them than even cars defined as "like new". The term "used" apparently means different things to different sellers.

```
price_cond <- split(vposts$price, vposts$condition)
boxplot(price_cond[c(22,13,7,10,8,42)], ylim=c(0,80000),
        ylab="Price", main="Price for 6 Most Frequent Ratings")
```



Here, "new", "like new", and "excellent" cars sell for essentially the same price, with "excellent" having more large outliers to broaden its distribution. "Good" and especially "fair" rated cars typically sell for much less, with the "fair" category having a much smaller IQR than the rest - this indicates a more compact distribution. The "used" category is as strange as it was in the last plot, with its distribution seemingly normal with only minimal right skewing.

```
age_cond <- split(2016-vposts$year, vposts$condition)
boxplot(age_cond[c(22,13,7,10,8,42)], ylim=c(0,90),
        ylab="Age", main="Age for 6 Most Frequent Ratings")
```



In general, as the car's age increases, it is more likely to be rated worse. All categories except for "new" and "used" have a very compact IQR relative to its spread. The "used" category continues to be a poorly defined term; the median and mean for age are both in the single digits and no used car in the dataset is older than about 25 years. Some or most of these cars should be examined and redefined.