STA141 Assignment 1: Part 2

Chad Pickering A03

Friday, October 09, 2015

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

Resources: Office hours, Piazza forums, R help documentation

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

1. THREE ANOMALIES:

ANOMALY 1:

Anomalies caused by human error in data entry in price, age, and odometer readings.

JUSTIFICATION: These anomalies falsely exaggerate the distribution, spread, and center of the data. Their removal enables further analysis to be more accurate and precise; the conclusions drawn will be of much greater consequence. It will be much more representative of the larger population of vehicles in the U.S. for sale.

CORRECTION PROCEDURE: The corrections are as follows:

PRICE:

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

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

```
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; no evidence to an alternative price
#Row 7101 has the new reasonable maximum of 569500.
```

AGE:

```
vposts$age <- 2016 - vposts$year
which(vposts$age == min(vposts$age, na.rm=TRUE))
```

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

```
vposts[21975, "age"] <- 14 #-6 before, set to likely keystroke 2022 -> 2002
vposts[8417, "age"] <- 12 #2012 before, because year was "4", set to year 2004, took difference
which(vposts$age == max(vposts$age, na.rm=TRUE))
```

```
## [1] 27557 27901 27902 28058 28059 28100 28373
```

```r
vposts[27901, "age"] <- NA
vposts[27902, "age"] <- NA
vposts[28058, "age"] <- NA
vposts[28059, "age"] <- NA
vposts[28100, "age"] <- NA
vposts[28373, "age"] <- NA #27557 is the first post of seven
                            #that seem like duplicates; I remove the other six here.
```

ODOMETER:

```r
#means of odometer for each type (skew more obvious)
tapply(vposts$odometer, vposts$type, mean, na.rm=TRUE)
```

```
##         bus convertible       coupe   hatchback    mini-van     offroad
##   127012.57    86136.34    93895.37    82067.51   113324.27   127440.49
##       other      pickup       sedan         SUV       truck         van
##    75484.30   123327.42   289129.01    99163.46   122799.24   106554.48
##       wagon
##   103657.57
```

```r
#medians for each type (normal center for comp.)
tapply(vposts$odometer, vposts$type, median, na.rm=TRUE)
```

```
##         bus convertible       coupe   hatchback    mini-van     offroad
##    123500.0     80650.0     83220.0     72600.0    115000.0    117000.0
##       other      pickup       sedan         SUV       truck         van
##     59720.5    117449.0     89867.5    100676.0    114136.5    107000.0
##       wagon
##    102004.0
```

```r
#all 5 of these odometer readings were not reasonable, no alt. to NA
head(sort(vposts$odometer, decreasing=TRUE), 5)
```

```
## [1] 1234567890   99999999   16000000   16000000    9500000
```

```r
which(vposts$odometer == 1234567890)
```

```
## [1] 18161
```

```r
vposts[18161, "odometer"] <- NA
which(vposts$odometer == 99999999)
```

```
## [1] 4530
```

```r
vposts[4530, "odometer"] <- NA
which(vposts$odometer == 16000000)
```

```
## [1] 19227 19537
```

```r
vposts[19227, "odometer"] <- NA
which(vposts$odometer == 16000000)
```

```
## [1] 19537
```

```r
vposts[19537, "odometer"] <- NA
which(vposts$odometer == 9500000)
```

```
## [1] 2741
```

```r
vposts[2741, "odometer"] <- NA

#removal of outliers result in adjusted means
tapply(vposts$odometer, vposts$type, mean, na.rm=TRUE)
```

```
##          bus convertible       coupe   hatchback    mini-van     offroad
##    127012.57    86136.34    93895.37    82067.51   113324.27   127440.49
##        other      pickup       sedan         SUV       truck         van
##     75484.30   123327.42    91042.22    99163.46   122799.24   106554.48
##        wagon
##    103657.57
```

```r
#medians and means are now much closer for many
tapply(vposts$odometer, vposts$type, median, na.rm=TRUE)
```

```
##          bus convertible       coupe   hatchback    mini-van     offroad
##     123500.0     80650.0     83220.0     72600.0    115000.0    117000.0
##        other      pickup       sedan         SUV       truck         van
##      59720.5    117449.0     89865.0    100676.0    114136.5    107000.0
##        wagon
##     102004.0
```

IMPACT ON ANALYZING THE DATA? Now that these few are cleaned (in reality, there would be a greater effort to go through more data if time was not a factor), more of the true distribution is gradually manifesting. This is shown in the tables I inserted before and after the odometer cleaning - most "type" means adjusted to closer to the corresponding median values since the amount of skewedness was reduced. Overall, evidence from the other columns led me to either adjust the value to a new value or to an NA value.

ANOMALY 2:

Anomalies caused by judgement errors in rating the vehicle's condition.

JUSTIFICATION: Errors in judgement are unavoidable, especially with so many individual users in the dataset. And it is always worse when a category is a rating; every person asked will have a different standard and definition for each category, like "used" or "excellent". To be absolutely certain that these categorical levels are of statistical significance, a third party would have to rate the condition of all of the vehicles in the database to maintain consistency and remove bias.

CORRECTION PROCEDURE: To re-organize the data, I examined each level with only one or few frequencies and reassigned them to a more appropriate larger level based on price, odometer, description, and whatever factors held clear evidence.

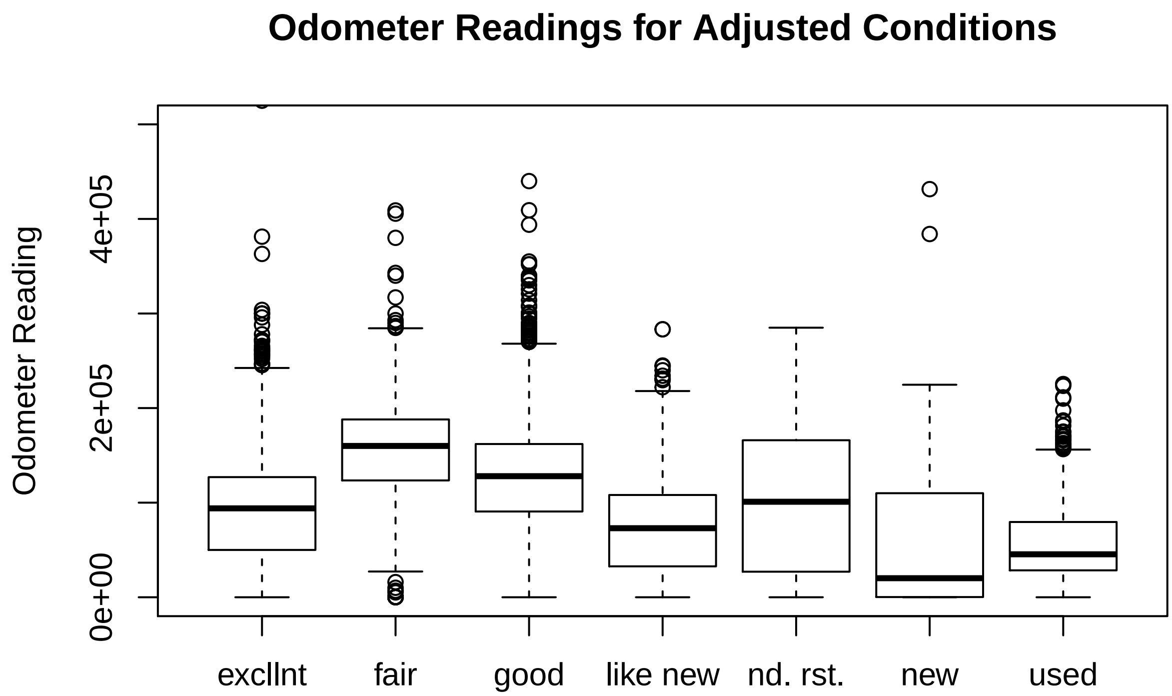
```
comb_condition <- vposts$condition
excellent <- c("excellent", "superb original", "very good")
used <- c("used", "preowned", "carfax guarantee!!", "pre-owned", "0used",
          "complete parts car, blown engine", "front side damage", "hit and run :( gently",
          "honnda", "mint", "pre owned", "preownes", "rough but runs", "certified",
          "muscle car restore", "nice rolling restoration", "restoration", "restore", "restored")
needs_restore <- c("needs bodywork", "project", "needs restoration!", "needs total restore",
                   "needs work", "not running", "salvage", "rebuildable project",
                   "restoration project", "needs work/for parts", "needs restored",
                   "needs restore", "project car", "parts")
good <- c("good", "207,400", "ac/heater", "nice", "nice teuck")
comb_condition <- as.character(vposts$condition)
upd_excellent <- comb_condition %in% excellent
upd_used <- comb_condition %in% used
upd_needsrestore <- comb_condition %in% needs_restore
upd_good <- comb_condition %in% good
comb_condition[upd_excellent] <- "excllnt"
comb_condition[upd_used] <- "used"
comb_condition[upd_needsrestore] <- "nd. rst."
comb_condition[upd_good] <- "good"
vposts$updcondition <- factor(comb_condition)
levels(vposts$updcondition)
```

```
## [1] "excllnt"  "fair"     "good"     "like new" "nd. rst." "new"
## [7] "used"
```

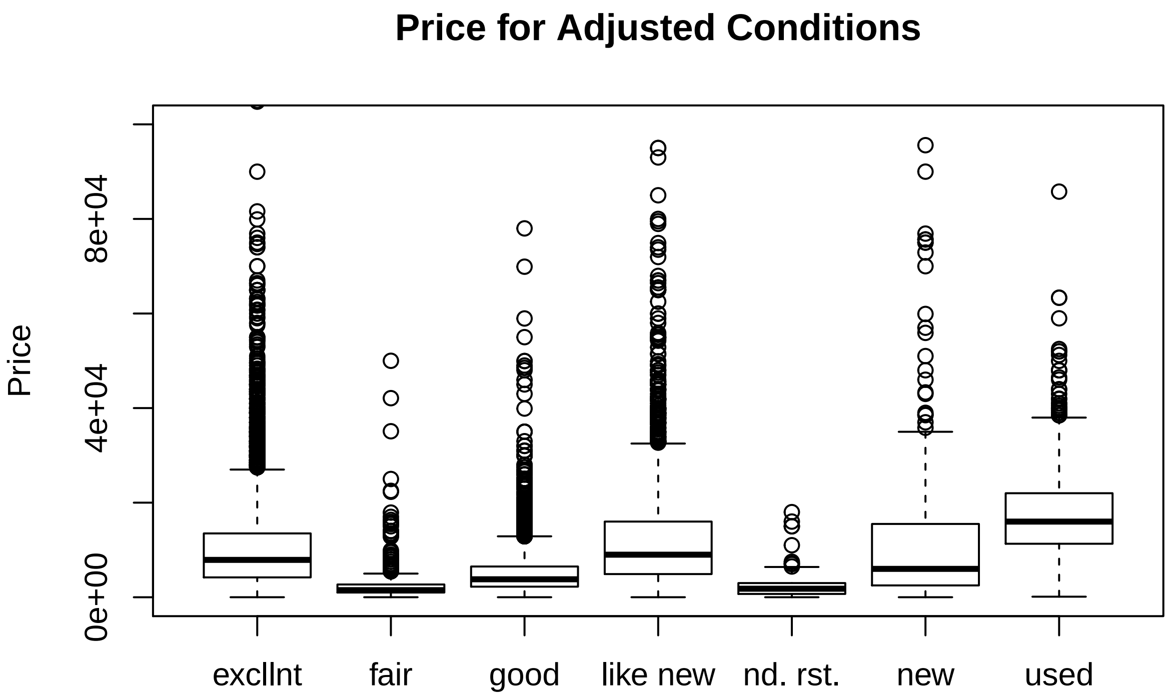
```
table(vposts$updcondition)
```

```
##
##  excllnt     fair     good like new nd. rst.      new     used
##     7555      776     4667     2898       82      273     1262
```

```
odo_newcond <- split(vposts$odometer, vposts$updcondition)
boxplot(odo_newcond, ylim=c(0,500000),
        ylab="Odometer Reading", main="Odometer Readings for Adjusted Conditions")
```




```r
price_newcond <- split(vposts$price, vposts$updcondition)
boxplot(price_newcond, ylim=c(0,100000),
        ylab="Price", main="Price for Adjusted Conditions")
```



IMPACT ON ANALYZING THE DATA? Combining categories promotes less clutter. In this way, comparing each level of condition with boxplots or some equivalent is much more straight-forward, as there are now only 7 categories to compare rather than 43. The benefit now is that instead of just analyzing the half dozen main categories that contain less than all of the data, any analysis now will include all of the data. The original database should not have had a field to fill out with whatever condition the user felt most correct; as we saw, typos and miscellaneous categories clutter things - a drop-down selection would be convenient for the user and the data scientist. As I said, though, a third-party rating system would be optimal.

ANOMALY #3:

Anomalies caused by lack of parsing OR human incompletion/brevity among “body” to other columns.

JUSTIFICATION, CORRECTION PROCEDURE, IMPACT: In some cases, information that is clearly given in the “body” column is not given in the appropriate column that is specific to that data, e.g. drive, transmission, etc. (Or this information is specific to the year/make of the vehicle but it is not given.) This is caused by parsing or using regular expressions incorrectly in the original database, or it is caused by the user not filling out all of the fields. A complete cleaning would make for a drastic change in the categorical data especially, and the relationships between them. With close to all available data gathered, the true proportions would be realized. Information can be gathered from the “body” column OR the specifications for the specific vehicle from an online database and applied to the appropriate columns currently NA/missing.

2. THREE INSIGHTS:

Interesting insight #1:

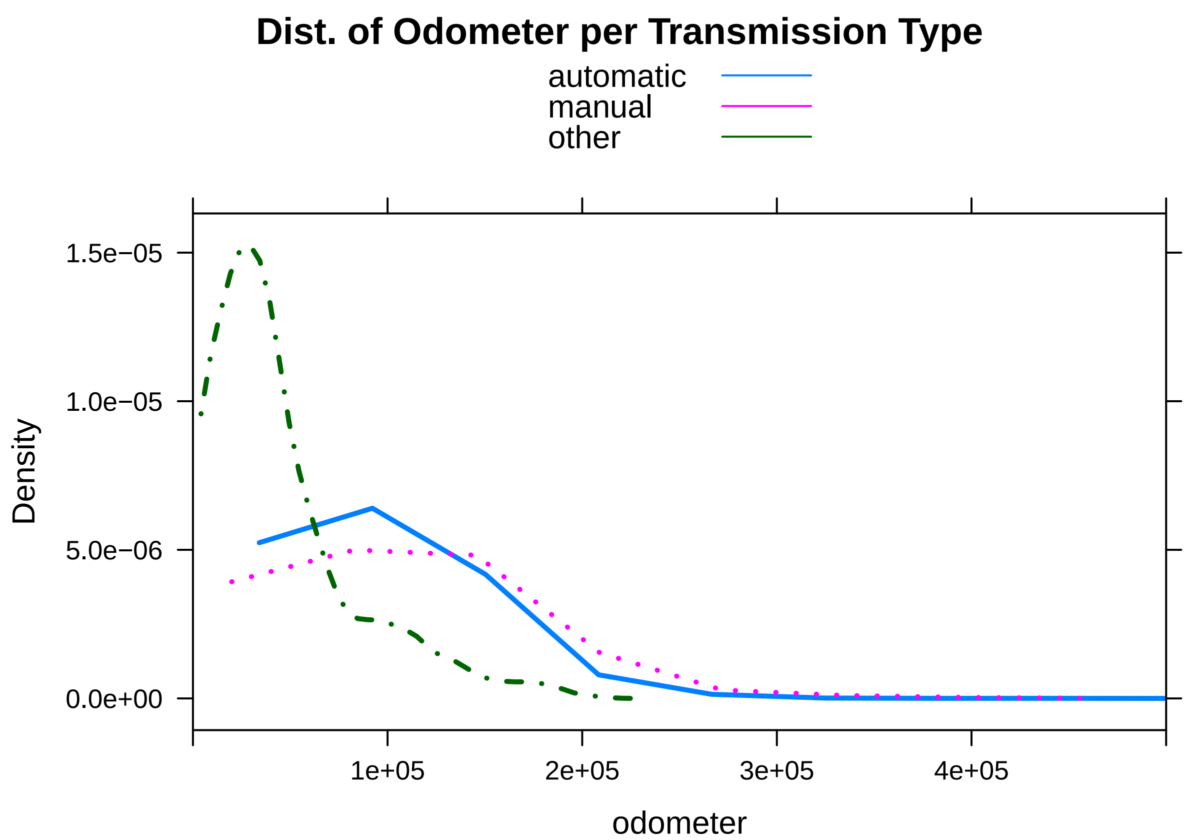
Interactions between odometer/transmission and age/transmission have an expected relationship besides the "other" category.

CONCLUSIONS: I constructed two density plots:

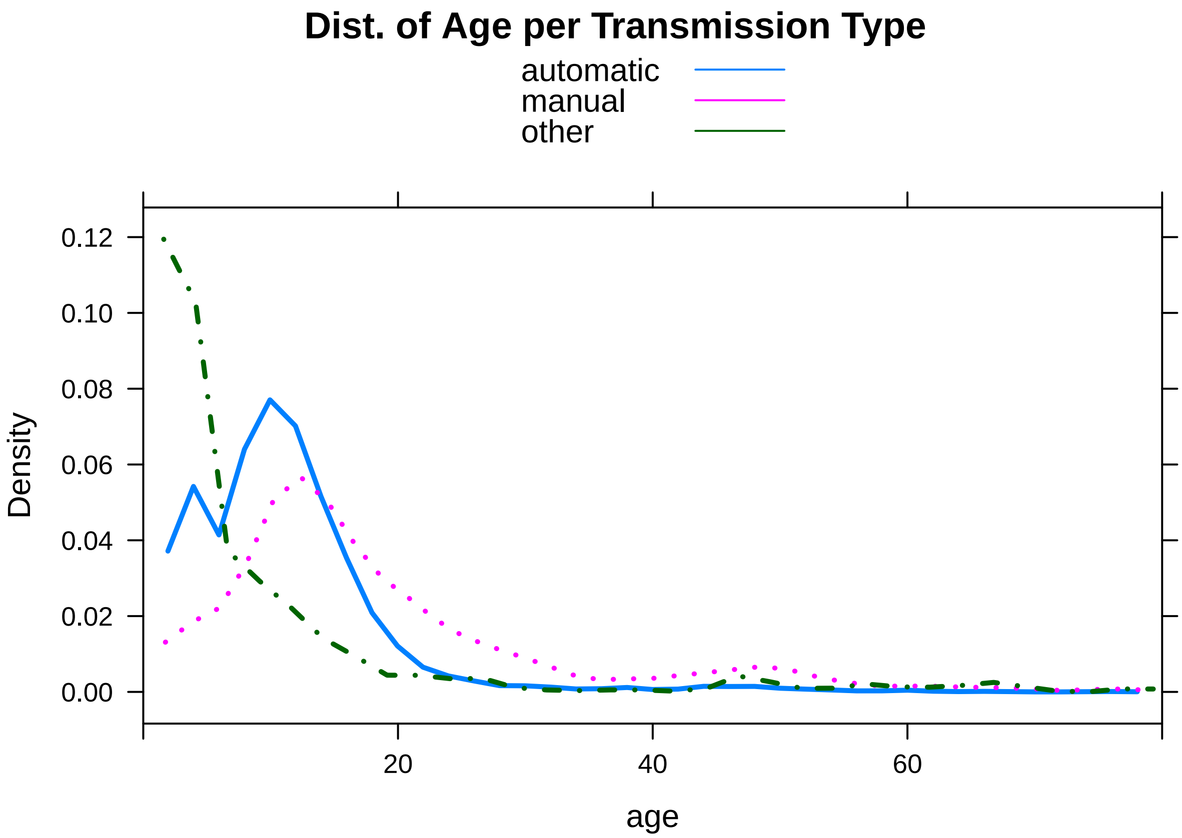
```
library(lattice)
```

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

```
densityplot( ~ odometer, vposts, group = transmission, plot.points=FALSE, xlim=c(0,500000),
             main="Dist. of Odometer per Transmission Type", lty=c(1, 3, 4), lwd=2.5, auto.key=TRUE)
```




```
densityplot( ~ age, vposts, group = transmission, plot.points=FALSE, xlim=c(0,80),
             main="Dist. of Age per Transmission Type", lty=c(1, 3, 4), lwd=2.5, auto.key=TRUE)
```



The automatic category has by far the most observations out of the three transmission types, and yet, the distributions for automatic and manual are roughly the same. Members of the "other" category have much less mileages than the other two categories, and ~90% of the cars are less than 10 years old - these observations agree. Similarly, manual vehicles have more miles on them, and it looks like ~90% of them are less than 35 years old, so a much larger spread than both automatic and other. In this way, we can see that age and odometer, through the scope of transmission types, are associated with each other.

Generalizable to other vehicle sales data? All of these observations/associations make logical sense without much evidence, so I would say that these observations and conclusions can be generalizable to other vehicles sales data contigent on large dataset size and location of interest. Information needed for further analysis include what constitutes an "other" transmission, and why there are so many of them; perhaps they are defined differently in other studies or datasets.

Interesting insight #2:

Patterns found in NA values.

CONCLUSIONS: When we find the column variables that have the same amount of NA values, we find that the same observations have NA values for the corresponding variables searched. This is proven here:

```
sum(is.na(vposts)) #total NAs in the data frame
```

```
## [1] 174397
```

```r
sapply(vposts, function(x) length(x[is.na(x)])) #number of NAs in each variable
```

```
##           id        title         body          lat         long
##            0            0            0        14445        14445
##       posted      updated        drive     odometer         type
##            0        15954        17276        10426        15892
##       header    condition    cylinders         fuel         size
##            0        17164        18864         2771        24985
## transmission      byOwner         city         time  description
##         1022            0            0            9            9
##     location          url        price         year        maker
##            9            9         3329            0          618
##  makerMethod          age updcondition
##            0            6        17164
```

```r
with(subset(vposts, lat == "NA" | long == "NA"), identical(lat, long))
```

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

```r
#the observations with lat and long missing are the same

with(subset(vposts, url == "NA" | description == "NA" | location == "NA"),
     identical(description, location))
```

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

```r
#the observations with description and location missing are the same

with(subset(vposts, url == "NA" | description == "NA" | location == "NA"),
     identical(url, description))
```

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

```r
#the observations with url and description missing are the same
```

Latitude and longitude are missing in the same observations, as well as description, location, and url, respectively.

Generalizable to other vehicle sales data? It is expected that if longitude data is missing, then latitude would be as well. But it is interesting that description, location, and url were all missing in only some observations simultaneously. This is not generalizable to any other dataset. It depends on each individual dataset: what the columns/variables of interest are, how they interact, and how accurately they are parsed.

Interesting insight #3:

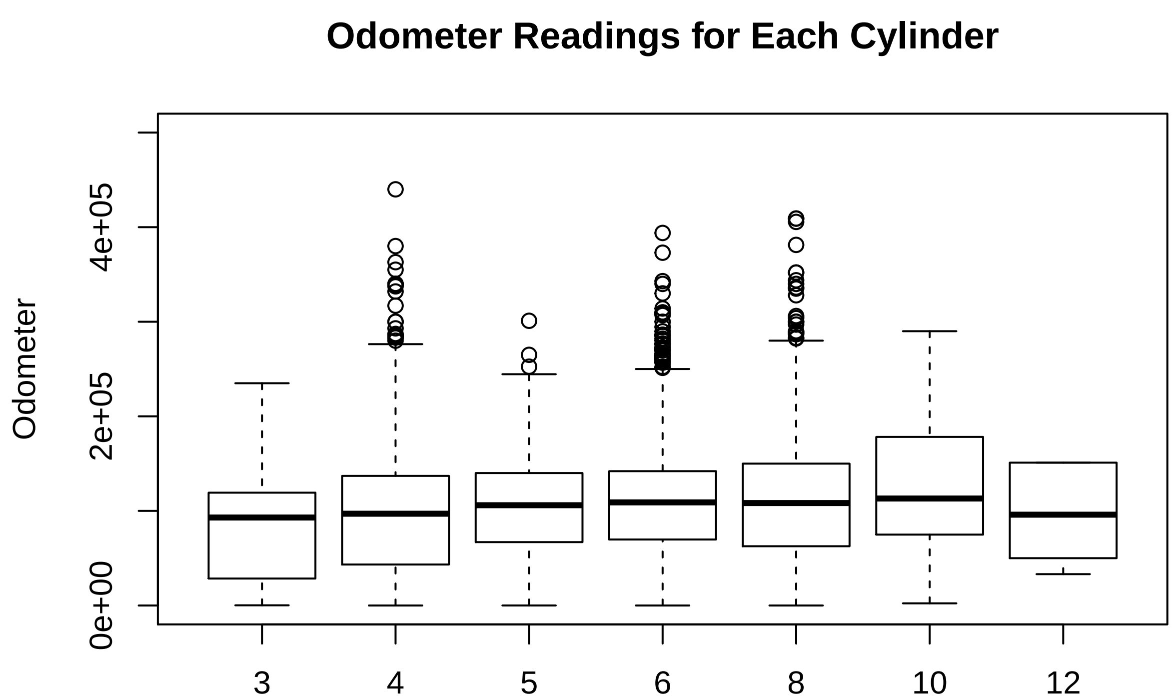
Relationships/interactions between cylinders and other columns/variables:

CONCLUSIONS: The following investigation is performed with the premise that I had no understanding that vehicles with odd-numbered cylinders existed. I examine several aspects of this subset:

```
unique(vposts$cylinders) #There are 3 and 5 cylinder vehicles??
```

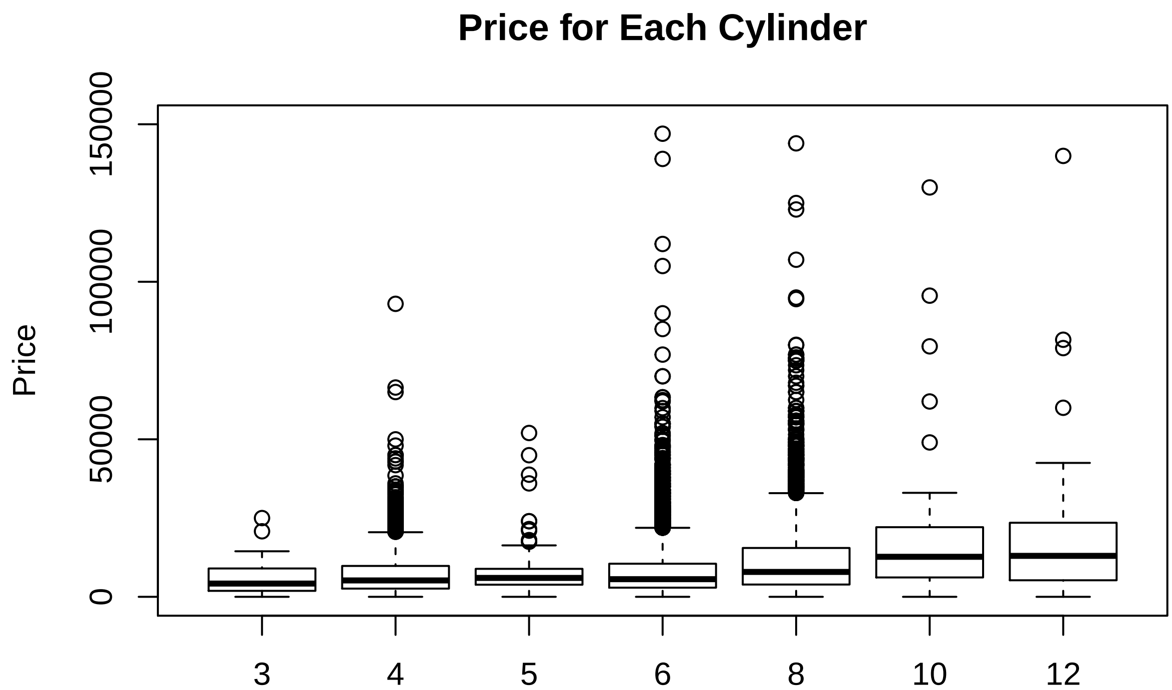
```
## [1] NA  6  4  8  5  3 12 10
```

```
cyl_odometer <- split(vposts$odometer, vposts$cylinders)
boxplot(cyl_odometer, ylim=c(0,500000), ylab="Odometer",
        main="Odometer Readings for Each Cylinder")
```



The odd-numbered cylinder types have roughly the same distribution amongst odometer.

```
cyl_price <- split(vposts$price, vposts$cylinders)
boxplot(cyl_price, ylim=c(0,150000), ylab="Price", main="Price for Each Cylinder")
```



The odd-numbered cylinder types have roughly the same distribution amongst price as 4 and 6 cylinder vehicles, with less outliers.

```
table(vposts$cylinders)
```

```
##
##    3    4    5    6    8   10   12
##   34 5519  223 6112 3844   53   28
```

There are 34 3-cylinder vehicles and an astounding 223 5-cylinder vehicles. There are less than 100 of 10-cylinder and 12-cylinder vehicles, but these are expected, as there are few semis and large trucks in the body of 34677 observations.

```
cyl <- as.factor(vposts$cylinders)
oddcylinders <- subset(vposts, (cyl == 3) | (cyl == 5)) #subset of 3 cyls or 5 cyls

sort(table(oddcylinders$city), decreasing=TRUE)
```

```
##
##    boston    denver     sfbay  lasvegas       sac   chicago       nyc
##        52        44        41        38        29        28        25
```

```
sort(table(vposts$city), decreasing=TRUE)
```

```
##
```

```
##      nyc   denver      sac lasvegas   boston    sfbay  chicago
##     4983     4979     4966     4963     4958     4942     4886
```

While city distribution among all vehicles is essentially uniform, between odd cylinders, it is skewed slightly. This is probably due to random error, and is insignificant.

```
head(sort(table(oddcylinders$maker), decreasing=TRUE), 8)
```

```
##
## volkswagen      volvo  chevrolet   mercedes       audi      honda
##         84         79         16         10          8          7
##      acura        geo
##          6          6
```

```
head(sort(table(vposts$maker), decreasing=TRUE), 24)
```

```
##
##       ford  chevrolet     toyota      honda     nissan      dodge
##       4266       3394       3332       2650       2473       1841
##        bmw   mercedes volkswagen       jeep    hyundai   chrysler
##       1657       1283       1116       1022        876        835
##      lexus      acura        gmc       audi   cadillac   infiniti
##        786        697        684        579        571        559
##      mazda     subaru    pontiac        kia      volvo      buick
##        550        531        431        395        371        363
```

Volkswagen and Volvo have a vast majority of observations within the odd cylindered vehicles, but the entire dataset has linearly decreasing frequencies among maker. This is relatively significant.

Upon simple boxplot analysis, both distributions from the subset are significantly less skewed than the original vposts dataset.

There is no need for correction! The concept that is prevalent here is that information contained in the dataset can be foreign to the data scientist at first, and the unexpected data merely needs a great deal of study. For example, I initially thought that the odd-numbered cylinders were inserted/entered accidentally. What I have done above is a sliver of the methodologies needed to fully understand the odd-numbered cylinders. I included this as a feature to drive home the point that an "anomaly" can be something absolutely unexpected to the data scientist, but is still perhaps not a traditional "outlier".

Generalizable to other vehicle sales data? With the understanding of any relationship or interaction between two variables comes a myriad of investigations to follow - how the two or more variables are associated, correlated, or vastly different. After understanding that vehicles having odd-numbered cylinders is just a very rare event, they can be treated only as such. We can expect roughly the same number of odd-numbered cylinders in similarly collected data.