Homework1

Lauren Stover

2/5/2021

Question 1: Gas Price Data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(rsample)  # for creating train/test splits
library(caret)
```

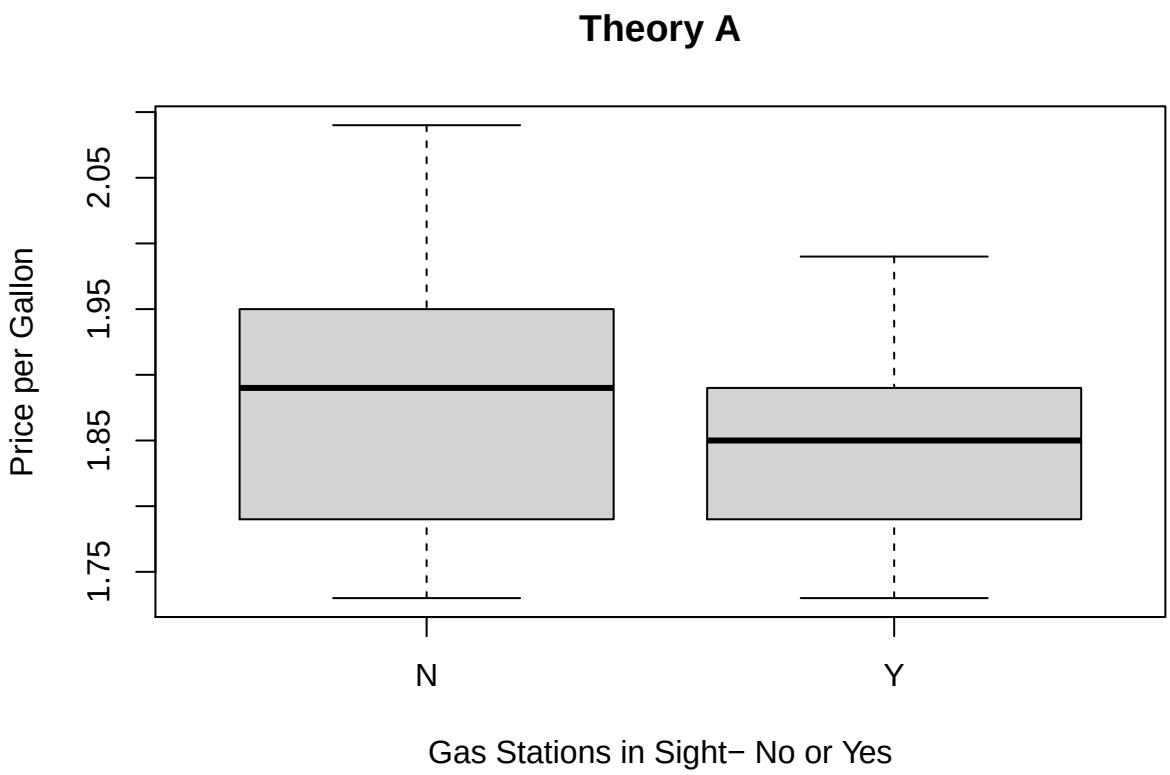
```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(modelr)
library(parallel)
library(foreach)
```

```
##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

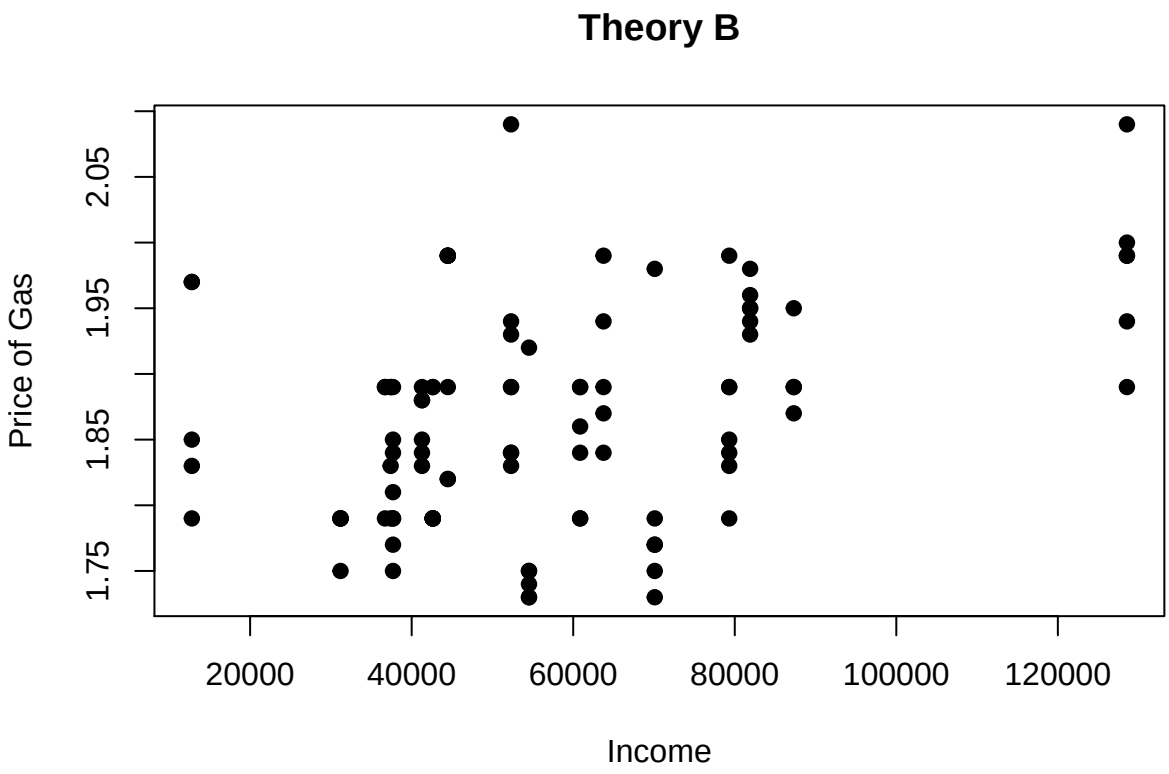
```
library(readxl)
```

```
GasPrices <- read_xls("GasPrices.xls")
boxplot(Price~Competitors,data=GasPrices, main="Theory A",
   xlab="Gas Stations in Sight- No or Yes", ylab="Price per Gallon")
```



Theory A states "Gas stations charge more if they lack direct competition in sight." From the boxplot above, the theory is supported from the data. We can clearly see the average and third quantile are higher for gas stations who do not have any direct competition in sight. The first quantiles and mimimums are roughly the same. The maximums though are drastically different in terms of prices. Those gas stations who have no direct competition in sight are charging more for gas.

```
attach(GasPrices)
plot(Income, Price, main="Theory B",
     xlab="Income", ylab="Price of Gas", pch=19)
```

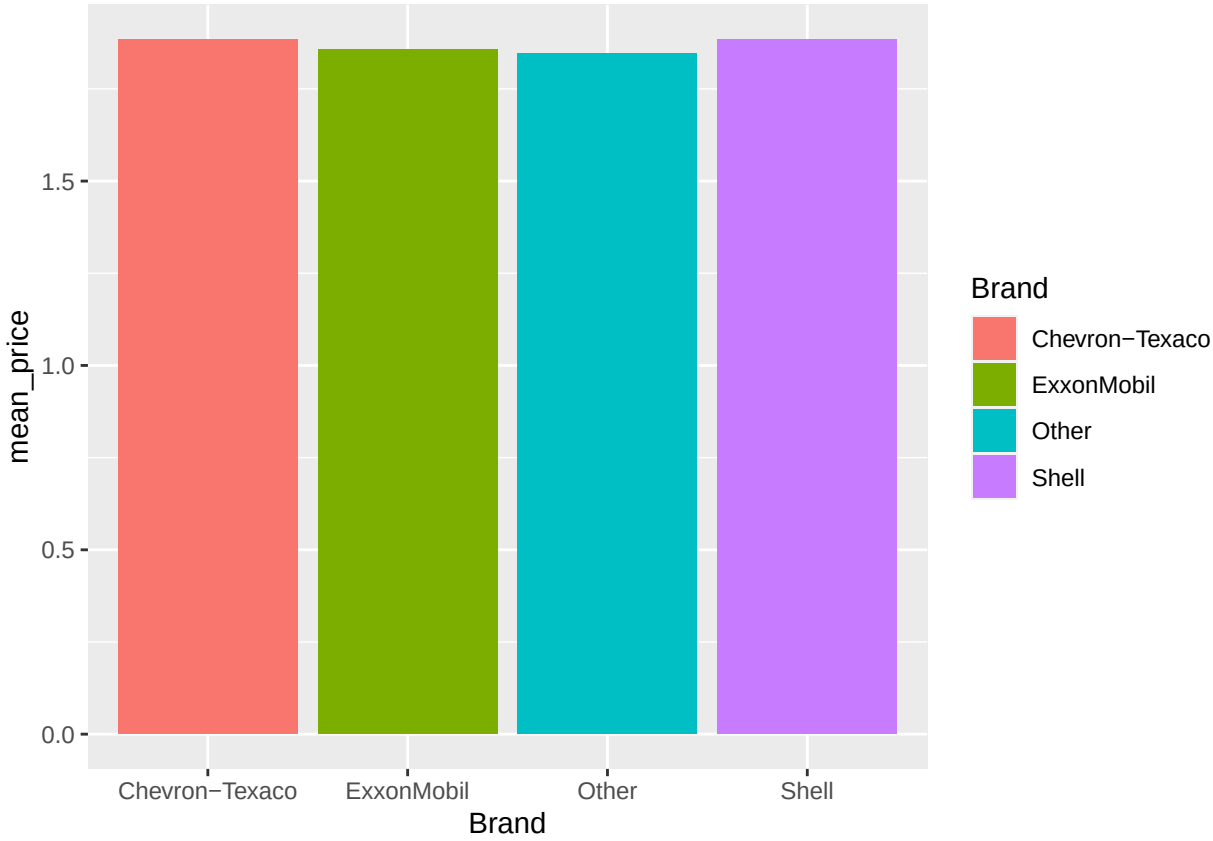


Theory B states "The richer the area, the higher the gas price." From the scatter plot above it would appear that it is true. As income of the area rises, gas prices are trending upwards. We can clearly see around areas of $80,000 of income and $120,000 of income that the starting prices of gas are much higher.

```
plot3=GasPrices %>%
  group_by(Brand) %>%
  summarize(mean_price=mean(Price))
```

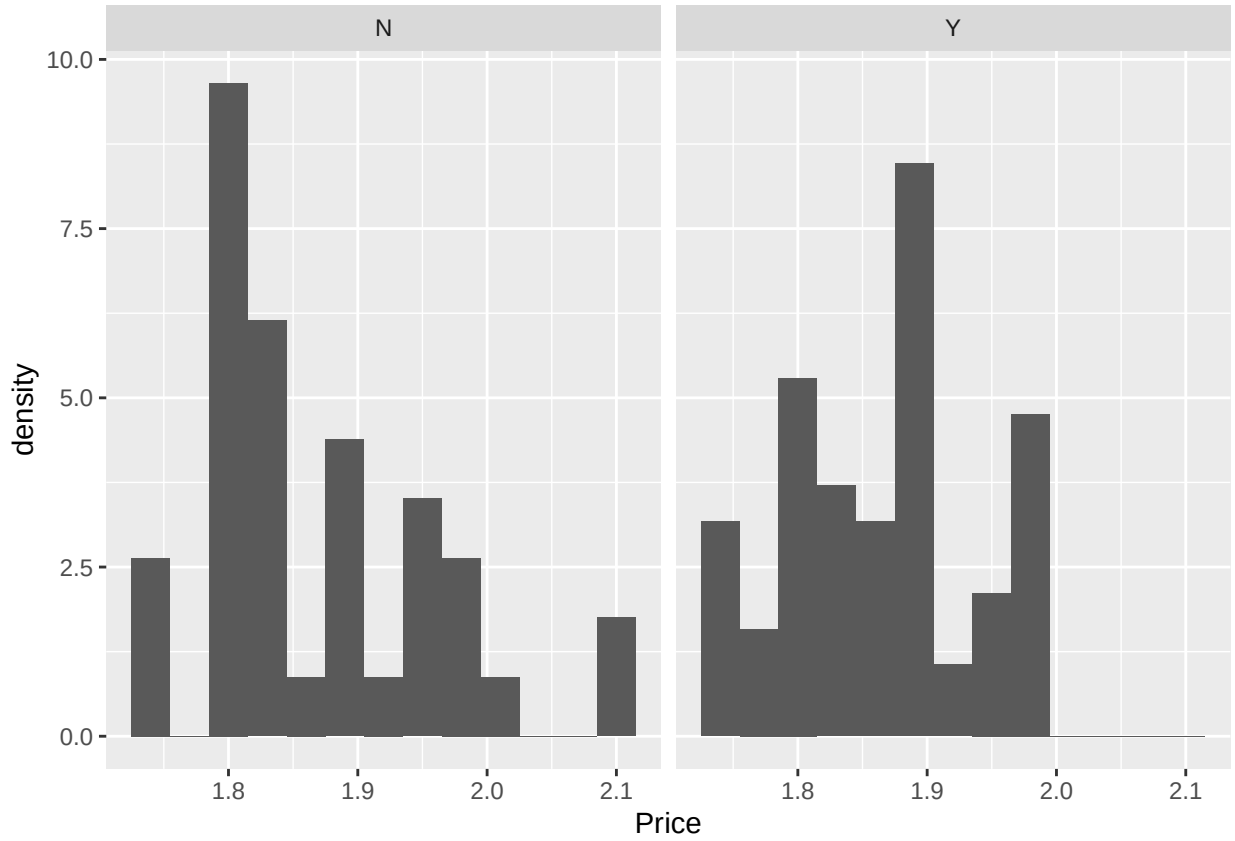
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(plot3)+
  geom_col(mapping= aes(x=Brand, y=mean_price, fill=Brand))
```



Theory C states that Shell charges more than other brands. From the graph one can observe that Shell charges very slightly more on average than the other brands, but the difference is almost unnoticeable. The average is at most probably a few cents difference per gallon, but in total that can add up.

```
ggplot(data=GasPrices)+
  geom_histogram(aes(x=Price, after_stat(density)),binwidth = 0.03)+
  facet_wrap(~Stoplight)
```

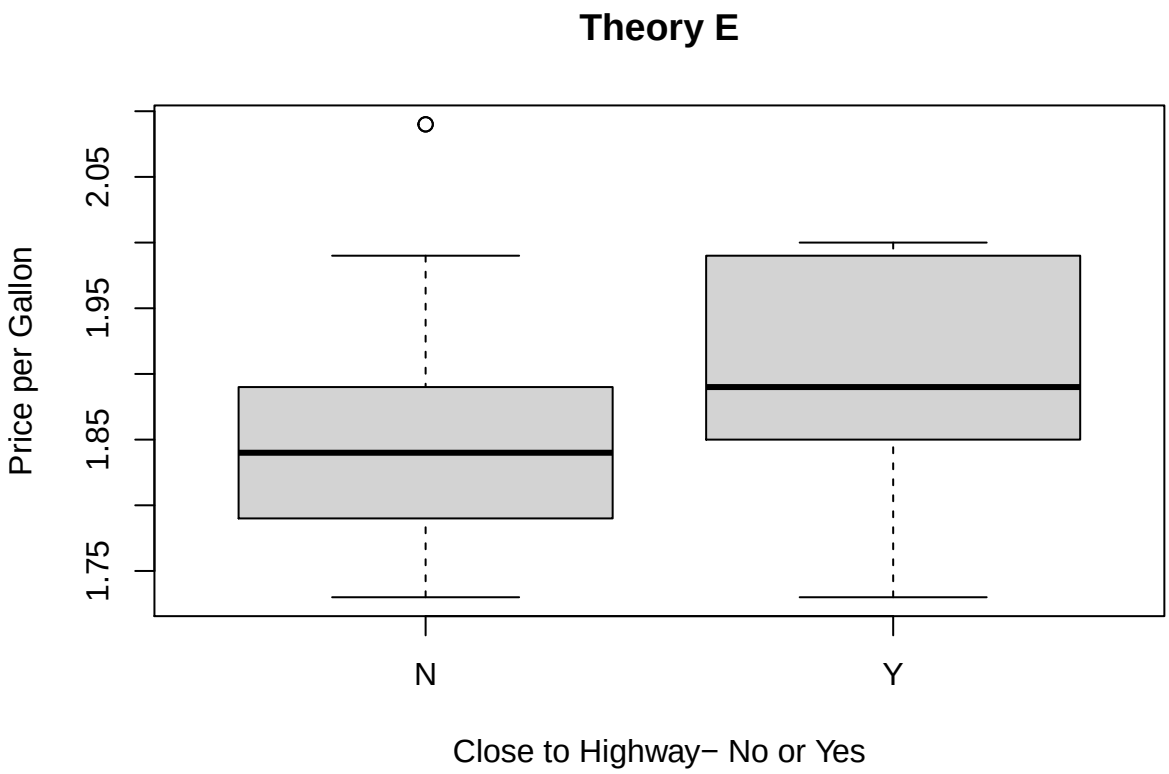


Theory D states that Gas stations at stoplights charge more. From this graph, you can see that gas stations near stoplights indeed do charge more. It appears that the price is more frequent around $1.90 near a stoplight and prices not near a stoplight are more concentrated around $1.80.

```
attach(GasPrices)
```

```
## The following objects are masked from GasPrices (pos = 3):
##
##     6, Address, Brand, CarWash, Competitors, Gasolines, Highway, ID,
##     Income, Interior, Intersection, IntersectionStoplight, Name, Price,
##     Pumps, Restaurant, Stoplight, Zipcode
```

```
boxplot(Price~Highway,data=GasPrices, main="Theory E",
   xlab="Close to Highway- No or Yes", ylab="Price per Gallon")
```



Theory E states "Gas stations with direct highway access charge more." As you can see by the plot above, the average price of gas for a gas station near a highway is higher than the average price of gas of a station not near a highway. The minimum and maximum values are actually quite close together though in price. As we can see, there is an outlier for prices not near a gas station. I would say though due to the average and the first and third quantiles, prices tend to be higher if a gas station is near a highway.

Question 2: Bike Share Data

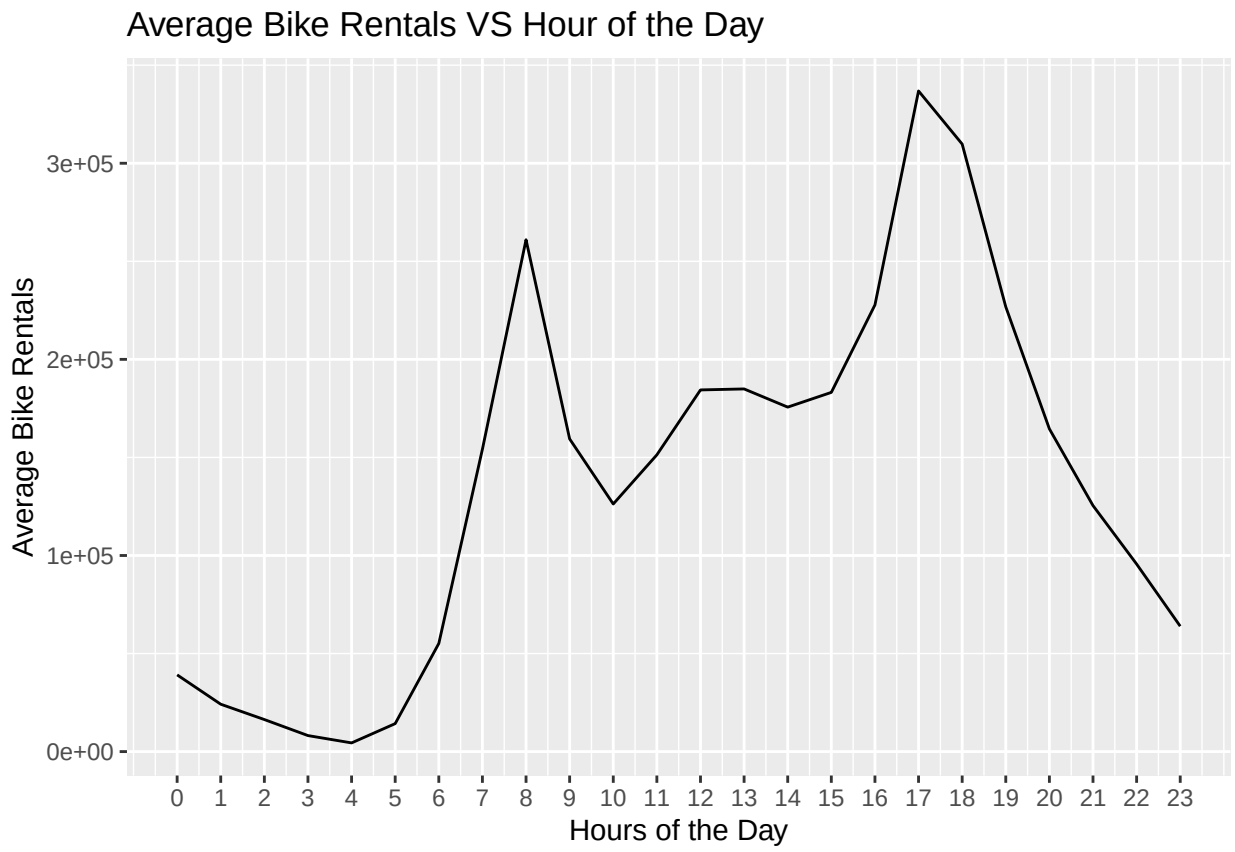
```
bikeshare <- read_xls("bikeshare.xls")
head(bikeshare)
```

```
## # A tibble: 6 x 12
##   instant dteday              season    yr  mnth    hr holiday weekday
##     <dbl> <dttm>               <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1       1 2011-01-01 00:00:00      1     0     1     0       0       6
## 2       2 2011-01-01 00:00:00      1     0     1     1       0       6
## 3       3 2011-01-01 00:00:00      1     0     1     2       0       6
## 4       4 2011-01-01 00:00:00      1     0     1     3       0       6
## 5       5 2011-01-01 00:00:00      1     0     1     4       0       6
## 6       6 2011-01-01 00:00:00      1     0     1     5       0       6
## # ... with 4 more variables: workingday <dbl>, weathersit <dbl>, temp <dbl>,
## #   total <dbl>
```

```
hours_total = bikeshare %>%
  group_by(hr) %>%
  summarize(rentals=sum(total))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
plot5 <- ggplot(hours_total) +
  geom_line(aes(x=hr, y=rentals)) +
  scale_x_continuous(breaks = 0:23)

plot5+ggtitle("Average Bike Rentals VS Hour of the Day")+xlab("Hours of the Day")+
  ylab("Average Bike Rentals")
```

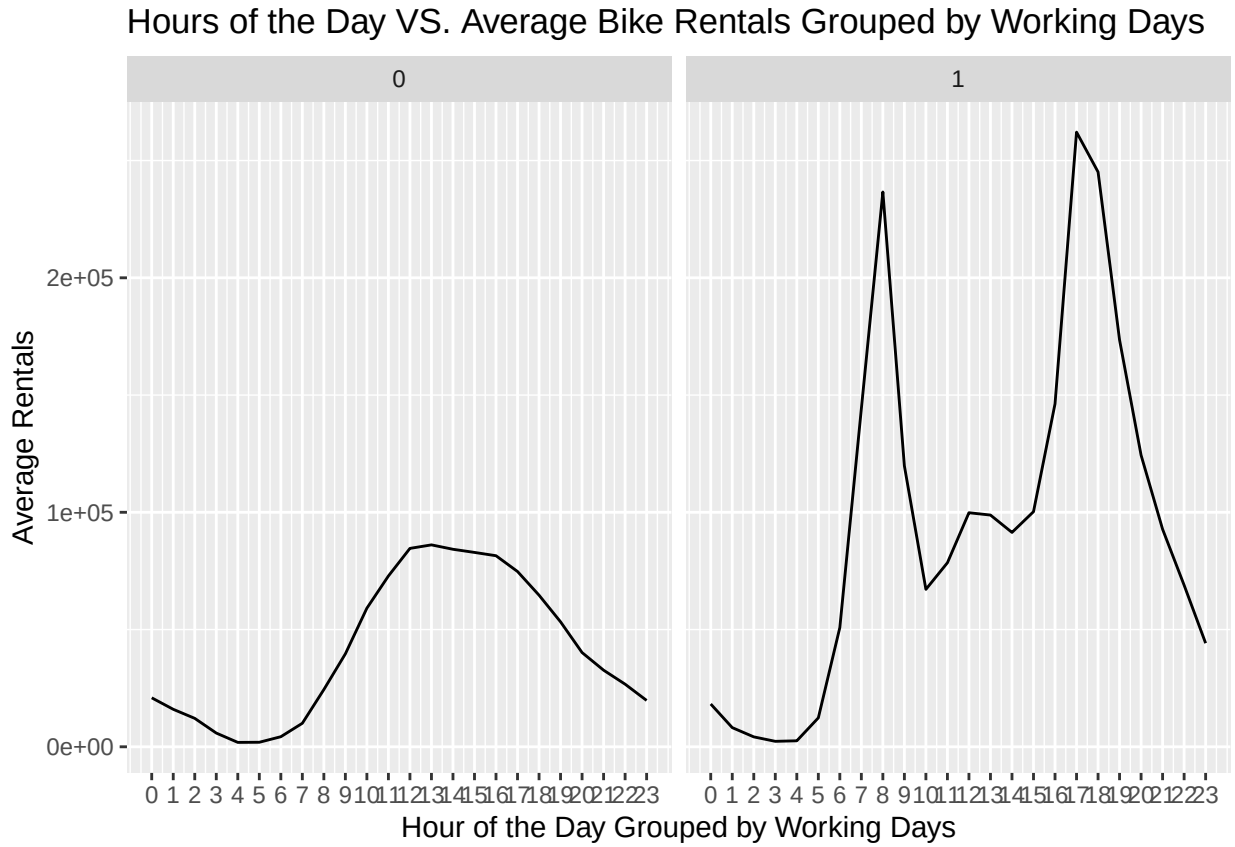


This line graph plots hours of the day vs the total rentals per hour of the day. We can see clearly during the hours when people are going to work and leaving work there is a surge in rentals.

```
q2p2 = bikeshare %>%
  group_by(hr, workingday) %>%
  summarize(rentals=sum(total))
```

```
## `summarise()` regrouping output by 'hr' (override with `.groups` argument)
```

```
plot3 <- ggplot(q2p2) +
  geom_line(aes(x=hr, y=rentals)) +
  scale_x_continuous(breaks = 0:23) +
  facet_wrap(~workingday)

plot3+ggtitle("Hours of the Day VS. Average Bike Rentals Grouped by Working Days") +
  xlab("Hour of the Day Grouped by Working Days") +ylab("Average Rentals")
```

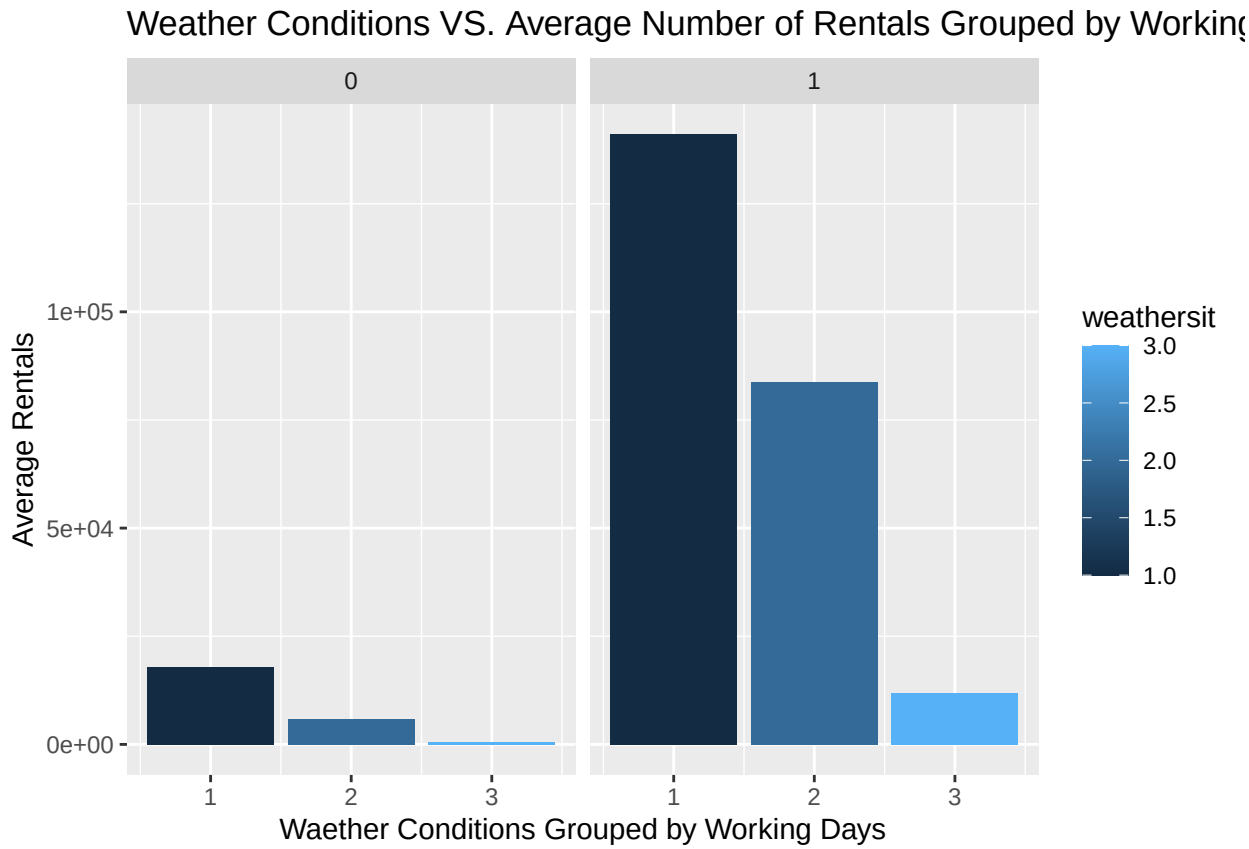


From this plot you can see that on working days, there is still a surge in the amount of rentals during the hours people travel to and from work. On non working days we can see that the surge in rentals is actually the opposite of working days. More people in general rent during the day as they are likely enjoying the city, going to and from activities, etc.

```
q2p3 = bikeshare %>%
  filter(hr==8)

q2p31=q2p3 %>%
  group_by(weathersit, workingday) %>%
  summarize(rentals=sum(total))
```

```
## `summarise()` regrouping output by 'weathersit' (override with `.groups` argument)
```

```
plot4 <-ggplot(q2p31)+
  geom_col(mapping= aes(x=weathersit, y=rentals, fill=weathersit)) +
  facet_wrap(~workingday)

plot4+ggtitle("Weather Conditions VS. Average Number of Rentals Grouped by Working Days") +
  xlab("Waether Conditions Grouped by Working Days") +ylab("Average Rentals")
```



From this plot we can see that during the 8 am hour there was no record of heavy rain, ice pallets, thunderstorm, mist, snow and fog. We can observe though that during working days and non working days people are most likely to rent during good weather, that is clear or a few clouds. On weather where this is a few clouds and mist on working days people are still likely to rent, but not nearly as much. on non working days when there is mist and clouds people are not renting bikes as much. On working and non working days when there is light snow or light rain the number of bike rentals severely drops.

Question 3 For this problem, I wanted to look into cancellations and various other data points. First, I looked into which days of the week had the highest cancellations.

```
ABIA <- read_xls("ABIA.xls")
```

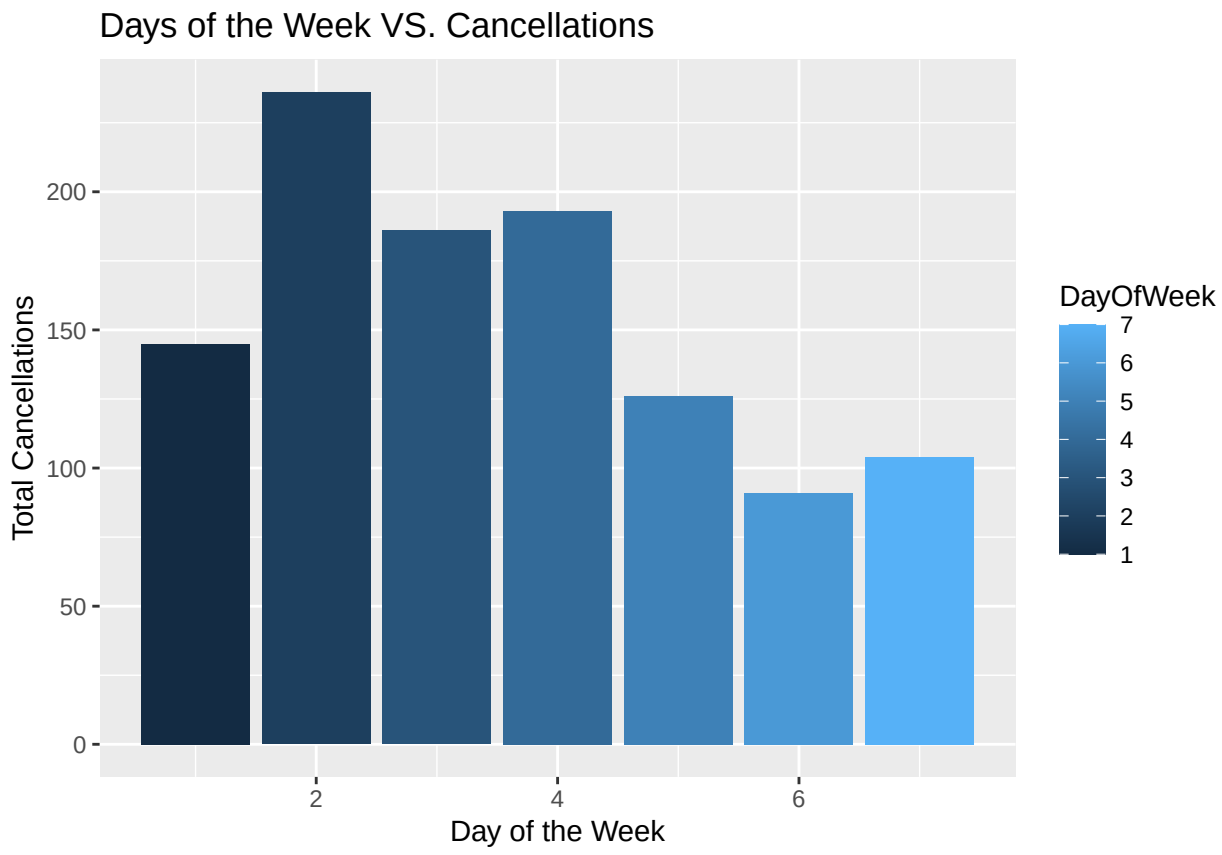
```
## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M7288 / R7288C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M8044 / R8044C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M8469 / R8469C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M11852 / R11852C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M28626 / R28626C13: got 'NA'
```

```
## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M51510 / R51510C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M54484 / R54484C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M65470 / R65470C13: got 'NA'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in M65501 / R65501C13: got 'NA'
```

```
q3p1=ABIA %>%
  filter(Cancelled==1)

helpme = q3p1 %>%
  group_by(DayOfWeek) %>%
  summarize(delays=sum(Cancelled))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
plot1 <- ggplot(helpme)+
  geom_col(mapping= aes(x=DayOfWeek, y=delays, fill=DayOfWeek))

plot1+ggtitle("Days of the Week VS. Cancellations") +
  xlab("Day of the Week") +ylab("Total Cancellations")
```



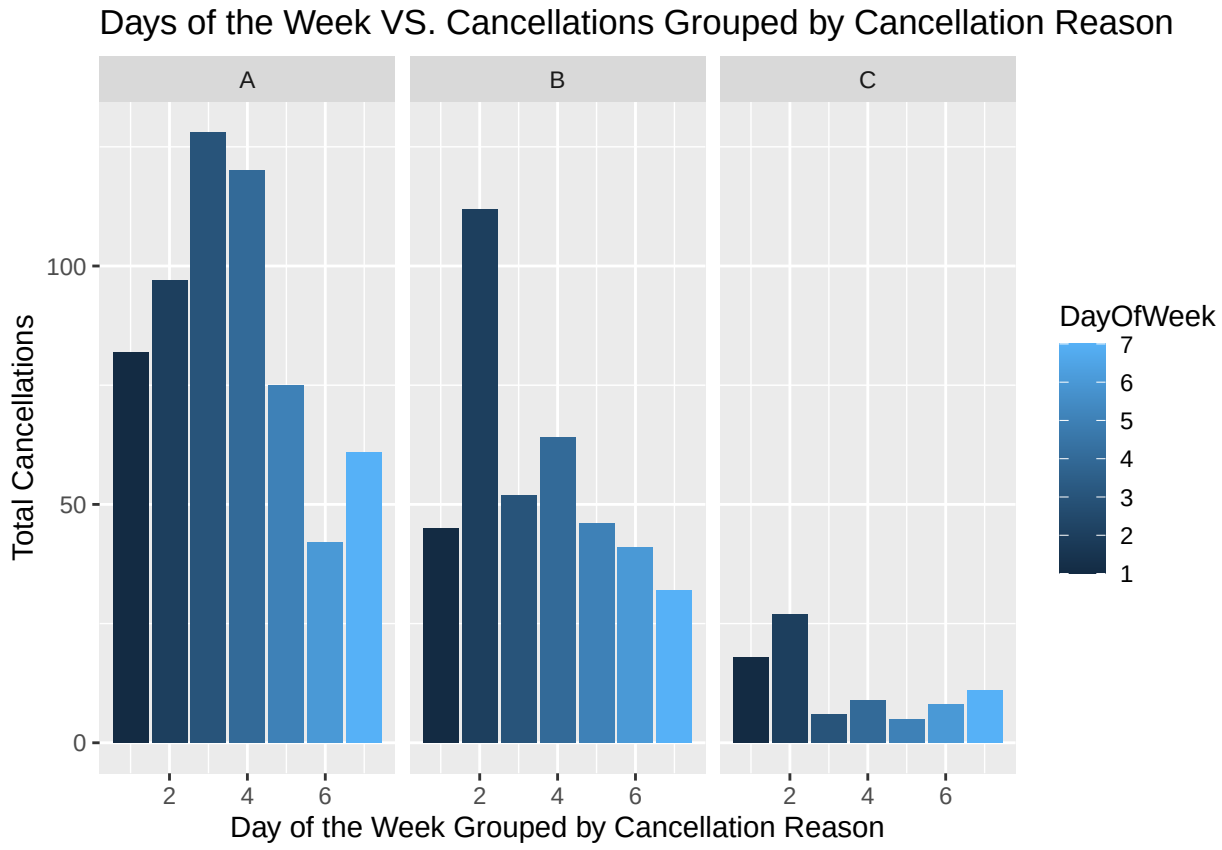
As you can see from the plot above, Tuesday flights actually have the highest number of cancellations. Saturday flights have the lowest number of cancellations.

Next, I wantedc to look at the various different reasons for cancellatins and days of the week they were most likely to frequent.

```
q3p12=q3p1 %>%
  group_by(CancellationCode, DayOfWeek) %>%
  summarize(delays=sum(Cancelled))
```

```
## `summarise()` regrouping output by 'CancellationCode' (override with `.groups` argument)
```

```
plot2 <- ggplot(q3p12)+
  geom_col(mapping= aes(x=DayOfWeek, y=delays, fill=DayOfWeek)) +
  facet_wrap(~CancellationCode)

plot2+ggtitle("Days of the Week VS. Cancellations Grouped by Cancellation Reason") +
  xlab("Day of the Week Grouped by Cancellation Reason") +ylab("Total Cancellations")
```



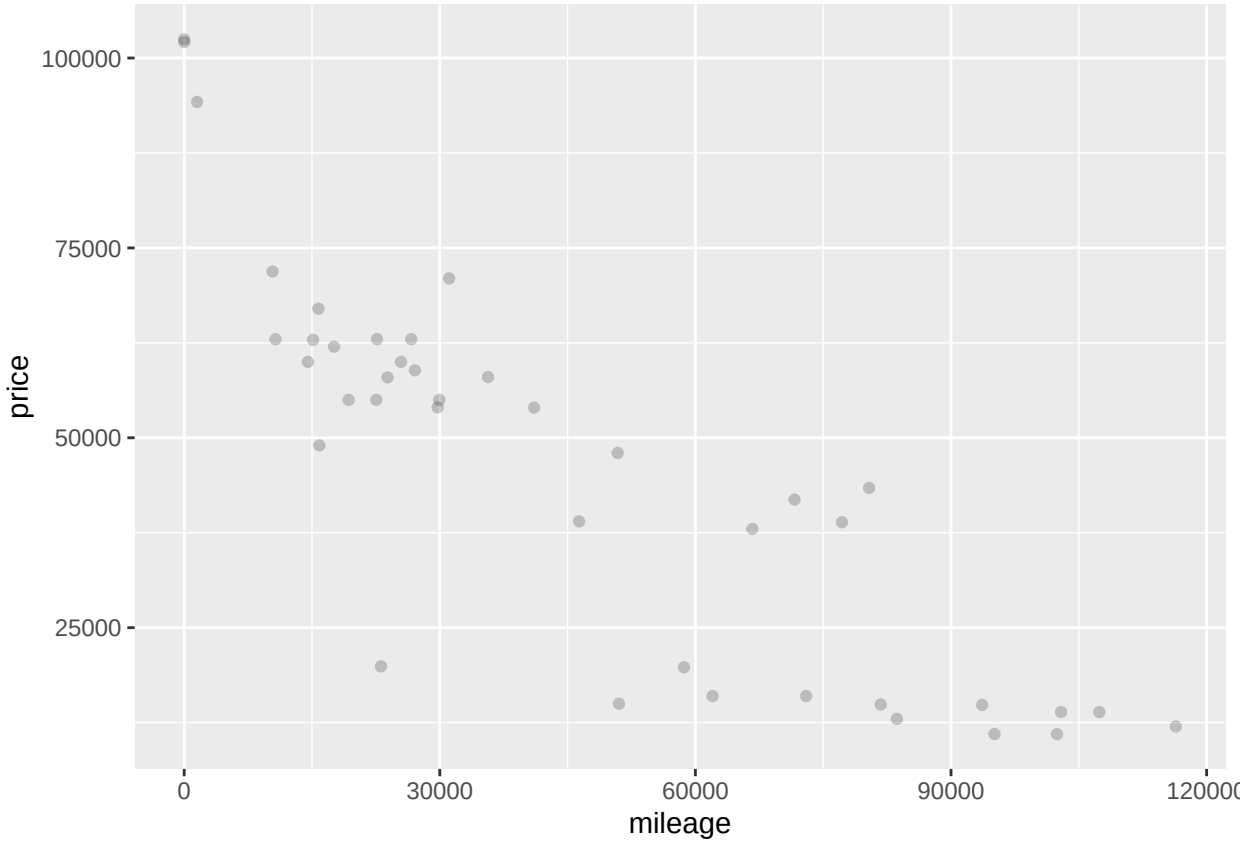
As you can see above, the number one reason flights are cancelled is because the carrier cancels the flight. The carrier is most likely to cancel the flight on a Wednesday than any other day. The carrier is less likely to cancel the flight on a Saturday. The next reason most flights are cancelled is because of weather. As you can see, on Tuesdays flights are most likely to be cancelled by weather and Sundays are the least likely day for this reason. Finally, NAS is the least common reason for cancellations. If an NAS cancellation were to happen it is most likely on a Tuesday and least likely on a Friday.

Question 4

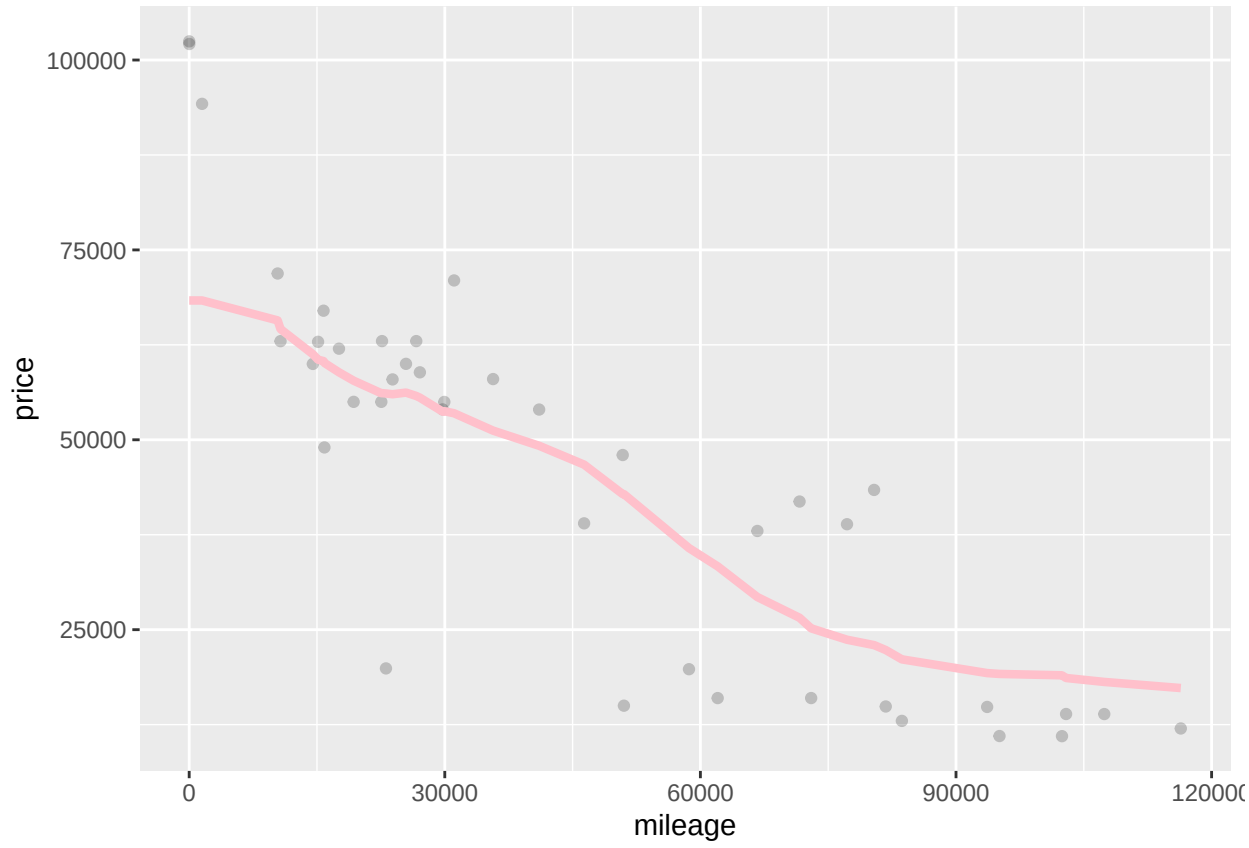
```
sclass<- read_xls("sclass.xls")
library(tidyverse)
library(ggplot2)
library(rsample)  # for creating train/test splits
library(caret)
library(modelr)
library(parallel)
library(foreach)

q4p1=sclass %>%
  filter(trim==350)
#getting just the models where the trim = 350

q4p1_split = initial_split(q4p1,prop=0.9)
q4p1_train=training(q4p1_split)
q4p1_test=testing(q4p1_split)
#creating the train and test split

trim1=lm(price~mileage, data=q4p1_train)
trim2=lm(price~poly(mileage,2),data=q4p1_train)
#linear and quadratic models

knn100=knnreg(price~mileage, data=q4p1_train, k=100)
rmse(knn100, q4p1_test)
```

```
## [1] 13814.52
```

```
#KNN with K=100

q4p1_test=q4p1_test %>%
  mutate(price_pred=predict(knn100, q4p1_test))

p_test=ggplot(data=q4p1_test) +
  geom_point(mapping=aes(x=mileage, y=price),alpha=0.2)

p_test
```




```
p_test+geom_line(aes(x=mileage, y=price_pred), color='pink', size=1.5)
```




```
#k-fold cross validation

K_folds=5

q4p1=q4p1%>%
  mutate(fold_id=rep(1:K_folds, length=nrow(q4p1)) %>% sample)

rmse_cv = foreach(fold = 1:K_folds, .combine='c') %do% {
  knn100 = knnreg(price ~ mileage,
                  data=filter(q4p1, fold_id != fold), k=100)
  modelr::rmse(knn100, data=filter(q4p1, fold_id == fold))
}

rmse_cv
```

```
## [1]  9235.101 14051.830 10489.691 10739.609 12036.123
```

```
#[1] 11615.32 10185.12 11053.70 11561.13 12431.35
mean(rmse_cv)  # mean CV error
```

```
## [1] 11310.47
```

```
#[1] 11369.32
sd(rmse_cv)/sqrt(K_folds)    # approximate standard error of CV error
```

```
## [1] 816.9561
```

```
#[1] 369.2668

q4p1_folds = crossv_kfold(q4p1, k=K_folds)

# map the model-fitting function over the training sets
models = map(q4p1_folds$train, ~ knnreg(price ~ mileage, k=100, data = ., use.all=FALSE))

errs = map2_dbl(models, q4p1_folds$test, modelr::rmse)


mean(errs)
```

```
## [1] 11321.63
```

```
#[1] 11295.14
sd(errs)/sqrt(K_folds)   # approximate standard error of CV error
```

```
## [1] 272.416
```

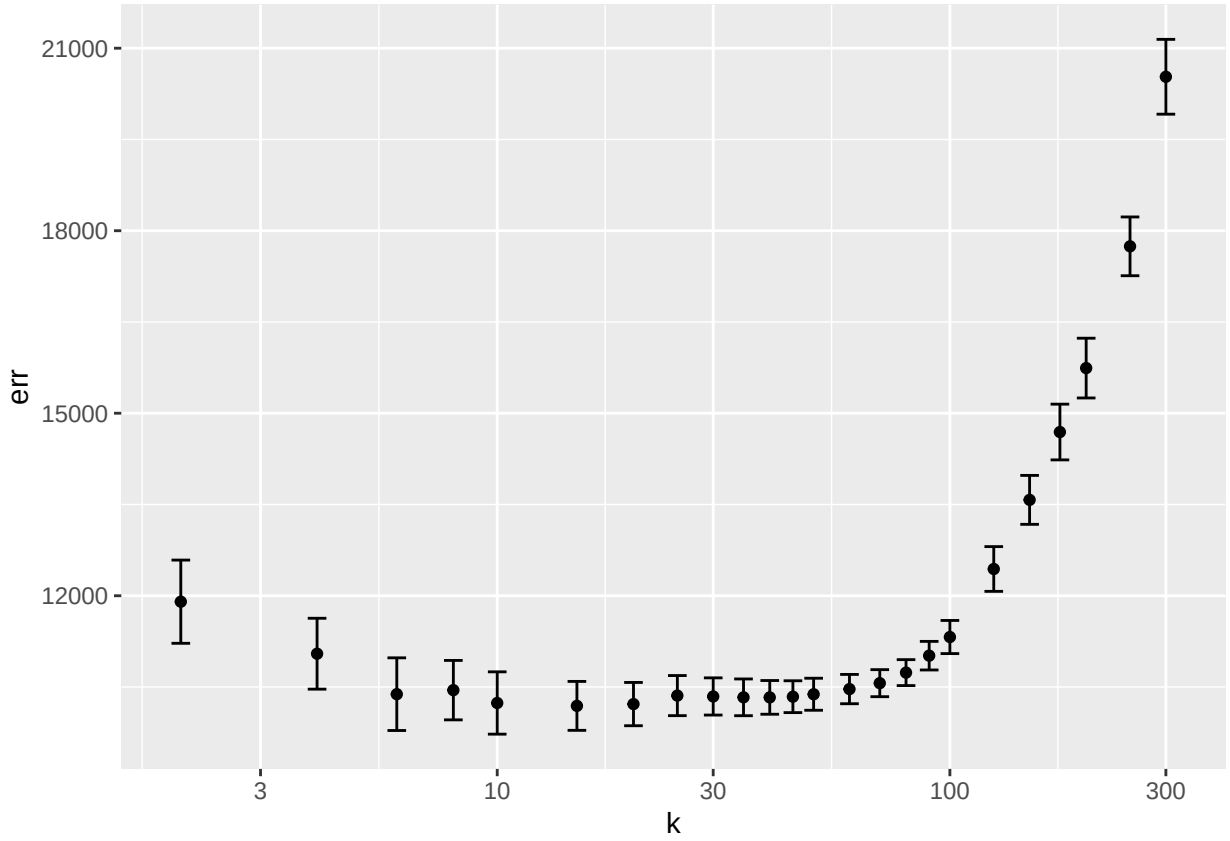
```
#[1] 391.3688

k_grid = c(2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45,
           50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 250, 300)

cv_grid = foreach(k = k_grid, .combine='rbind') %dopar% {
  models = map(q4p1_folds$train, ~ knnreg(price ~ mileage, k=k, data = ., use.all=FALSE))
  errs = map2_dbl(models, q4p1_folds$test, modelr::rmse)
  c(k=k, err = mean(errs), std_err = sd(errs)/sqrt(K_folds))
} %>% as.data.frame
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
head(cv_grid)
```

```
##           k      err  std_err
## result.1  2 11902.38 683.9798
## result.2  4 11047.42 582.4061
## result.3  6 10382.17 597.1688
## result.4  8 10448.39 488.5728
## result.5 10 10237.21 512.2077
## result.6 15 10189.87 402.2156
```

```
# plot means and std errors versus k
ggplot(cv_grid) +
  geom_point(aes(x=k, y=err)) +
  geom_errorbar(aes(x=k, ymin = err-std_err, ymax = err+std_err)) +
  scale_x_log10()
```




```
head(cv_grid)
```

```
##           k      err  std_err
## result.1  2 11902.38 683.9798
## result.2  4 11047.42 582.4061
## result.3  6 10382.17 597.1688
## result.4  8 10448.39 488.5728
## result.5 10 10237.21 512.2077
## result.6 15 10189.87 402.2156
```

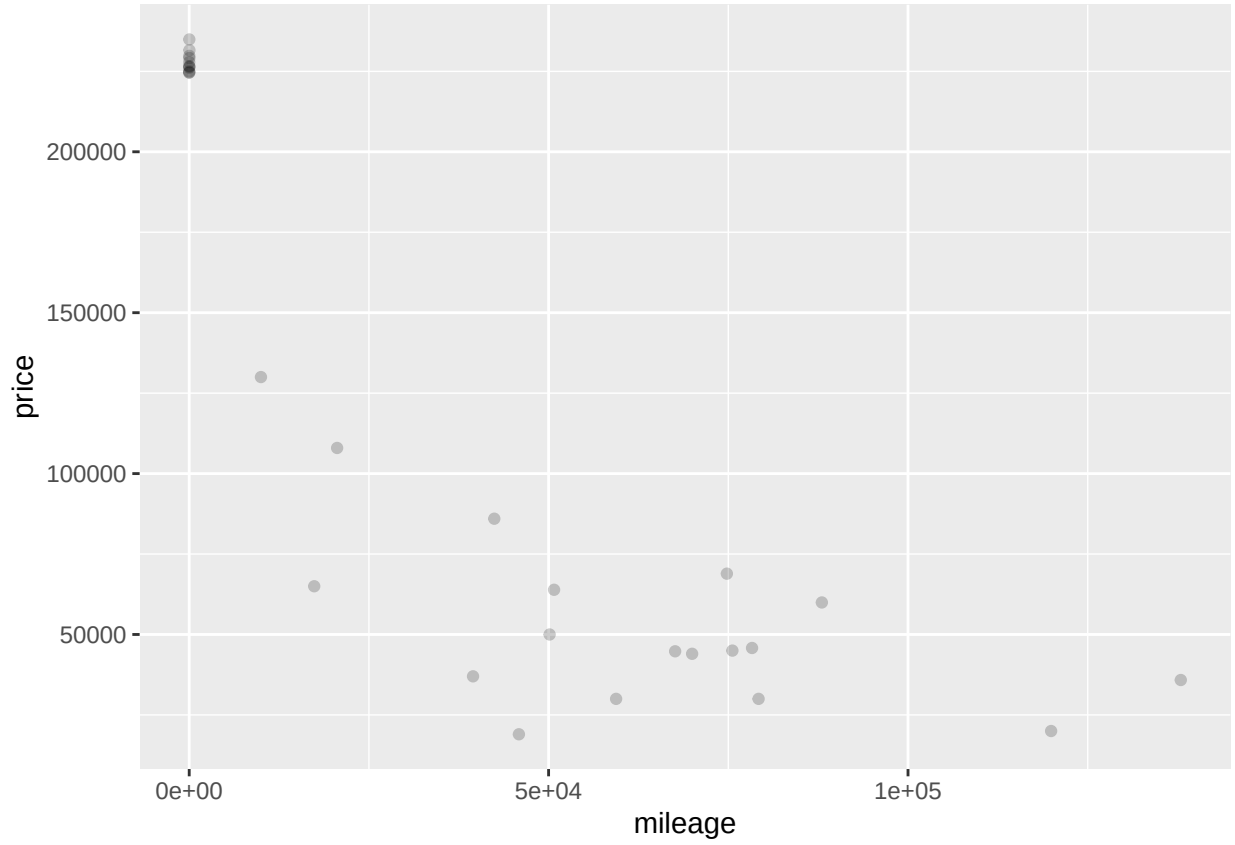
Now for the trim with 65 AMG

```
q4p2=sclass %>%
  filter(trim=='65 AMG')
#getting just the models where the trim = 350

q4p2_split = initial_split(q4p2,prop=0.9)
q4p2_train=training(q4p2_split)
q4p2_test=testing(q4p2_split)
#creating the train and test split

trim3=lm(price~mileage, data=q4p2_train)
trim4=lm(price~poly(mileage,2),data=q4p2_train)
#linear and quadratic models
```

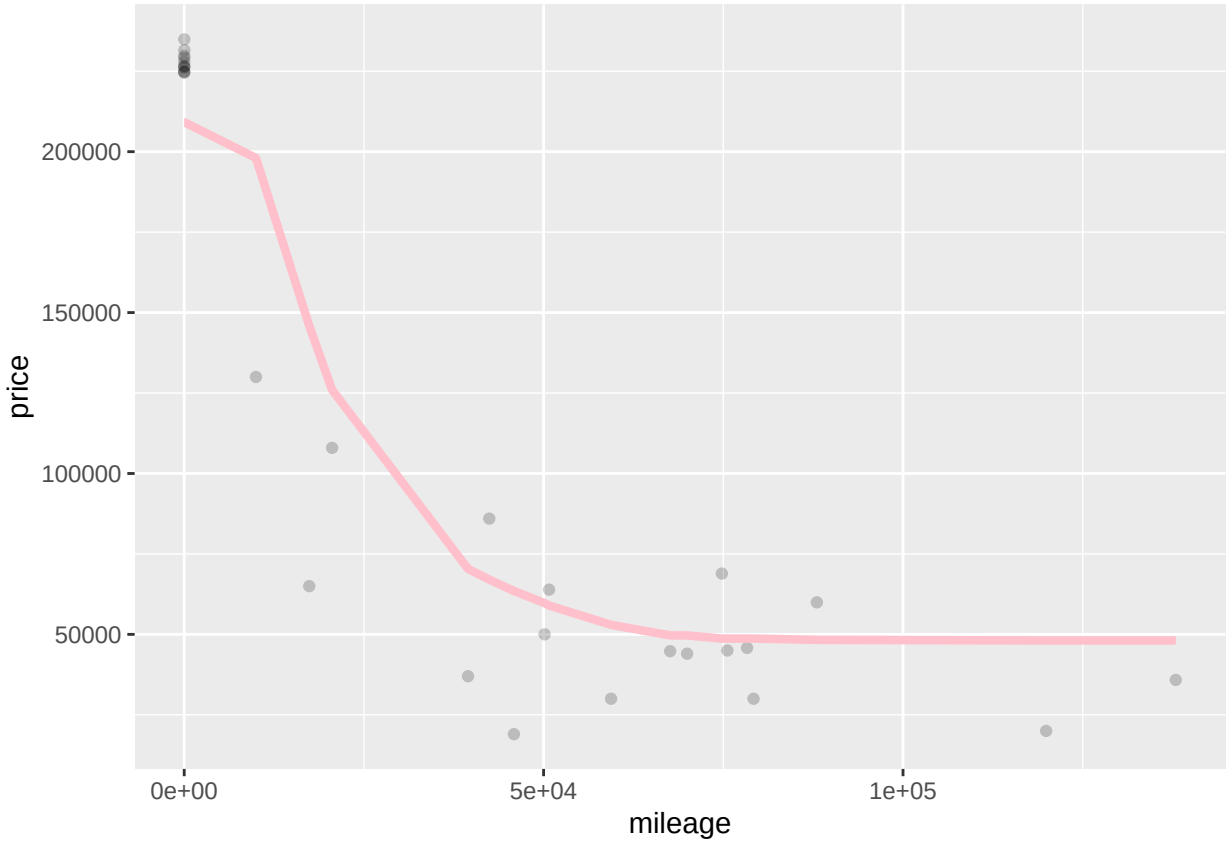
```
#knn WITH K=100
knn100=knnreg(price~mileage, data=q4p2_train, k=100)
rmse(knn100, q4p2_test)
```

```
## [1] 27218.21
```

```
#[1] 25279.13


q4p2_test=q4p2_test %>%
  mutate(price_pred1=predict(knn100, q4p2_test))

p_test1=ggplot(data=q4p2_test) +
  geom_point(mapping=aes(x=mileage, y=price),alpha=0.2)

p_test1
```




```
p_test1+geom_line(aes(x=mileage, y=price_pred1), color='pink', size=1.5)
```




```
#k-fold cross validation

K_folds=5

q4p2=q4p2%>%
  mutate(fold_id=rep(1:K_folds, length=nrow(q4p2)) %>% sample)

rmse_cv1 = foreach(fold = 1:K_folds, .combine='c') %do% {
  knn100 = knnreg(price ~ mileage,
                  data=filter(q4p2, fold_id != fold), k=100)
  modelr::rmse(knn100, data=filter(q4p2, fold_id == fold))
}

rmse_cv
```

```
## [1]  9235.101 14051.830 10489.691 10739.609 12036.123
```

```
# [1] 11615.32 10185.12 11053.70 11561.13 12431.35
mean(rmse_cv)  # mean CV error
```

```
## [1] 11310.47
```

```
#[1] 11369.32
sd(rmse_cv)/sqrt(K_folds)    # approximate standard error of CV error
```

```
## [1] 816.9561
```

```
#[1] 369.2668

q4p2_folds = crossv_kfold(q4p2, k=K_folds)

# map the model-fitting function over the training sets
models = map(q4p2_folds$train, ~ knnreg(price ~ mileage, k=100, data = ., use.all=FALSE))

errs = map2_dbl(models, q4p2_folds$test, modelr::rmse)


mean(errs)
```

```
## [1] 35550.31
```

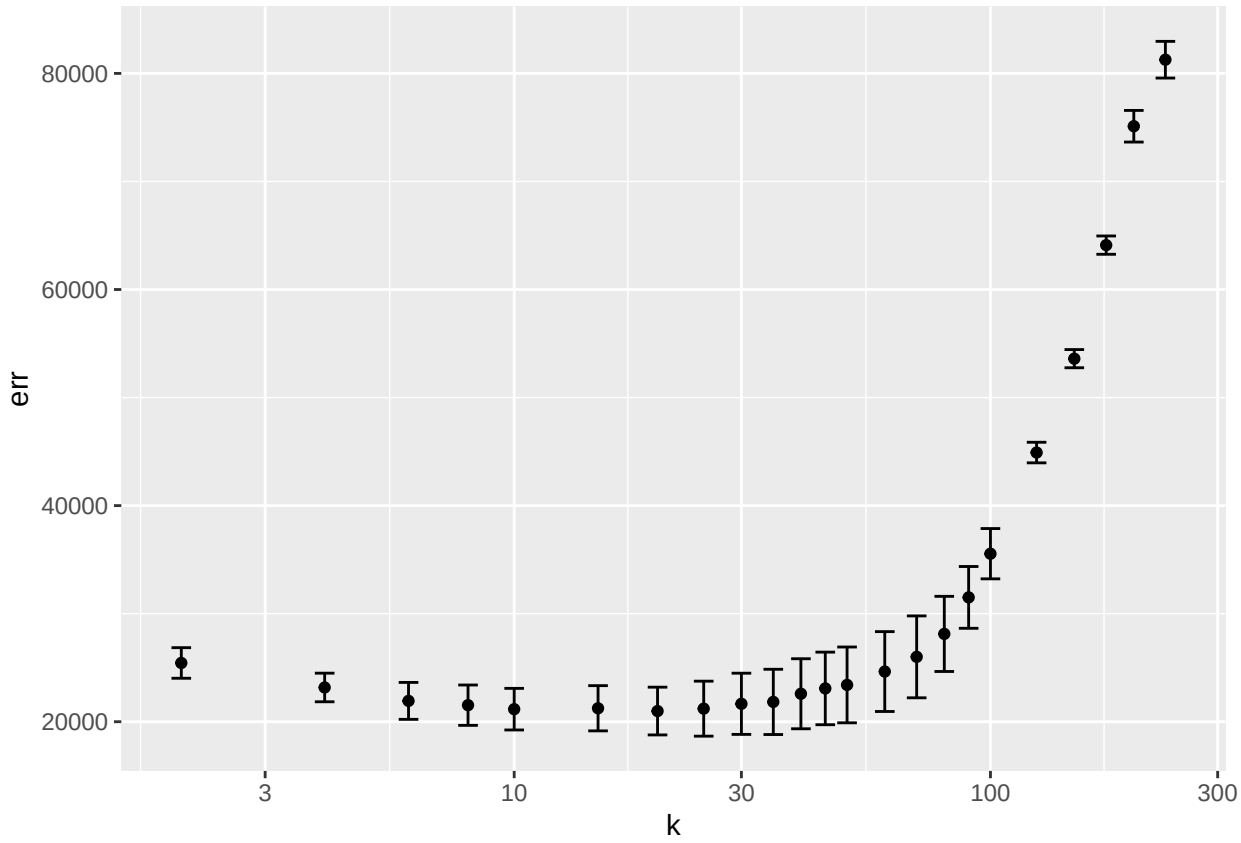
```
#[1] 35238.17
sd(errs)/sqrt(K_folds)   # approximate standard error of CV error
```

```
## [1] 2326.415
```

```
#[1] 1756.868

k_grid = c(2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45,
           50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 233)

cv_grid = foreach(k = k_grid, .combine='rbind') %dopar% {
  models = map(q4p2_folds$train, ~ knnreg(price ~ mileage, k=k, data = ., use.all=FALSE))
  errs = map2_dbl(models, q4p2_folds$test, modelr::rmse)
  c(k=k, err = mean(errs), std_err = sd(errs)/sqrt(K_folds))
} %>% as.data.frame

head(cv_grid)
```

```
##           k      err  std_err
## result.1  2 25438.85 1415.027
## result.2  4 23169.54 1324.187
## result.3  6 21928.66 1708.383
## result.4  8 21531.73 1866.108
## result.5 10 21162.15 1924.910
## result.6 15 21245.60 2093.042
```

```
# plot means and std errors versus k
ggplot(cv_grid) +
  geom_point(aes(x=k, y=err)) +
  geom_errorbar(aes(x=k, ymin = err-std_err, ymax = err+std_err)) +
  scale_x_log10()
```




```
head(cv_grid)
```

```
##          k      err  std_err
## result.1  2 25438.85 1415.027
## result.2  4 23169.54 1324.187
## result.3  6 21928.66 1708.383
## result.4  8 21531.73 1866.108
## result.5 10 21162.15 1924.910
## result.6 15 21245.60 2093.042
```

The 350 trim yields the larger optimal value of K. I believe this is because there is less variation within the trim model itself. The model has less mileage and the mileage was mostly concentrated under 50,000. In fact most of the 350 trim was concentrated around 25,000 miles. This would make the price prediction slightly easier.