Homework1

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

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

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.0     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)
```

```
## Warning: package 'rsample' was built under R version 4.0.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 4.0.3

##
## Attaching package: 'foreach'

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

```
library(modelr)
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 4.0.3
```

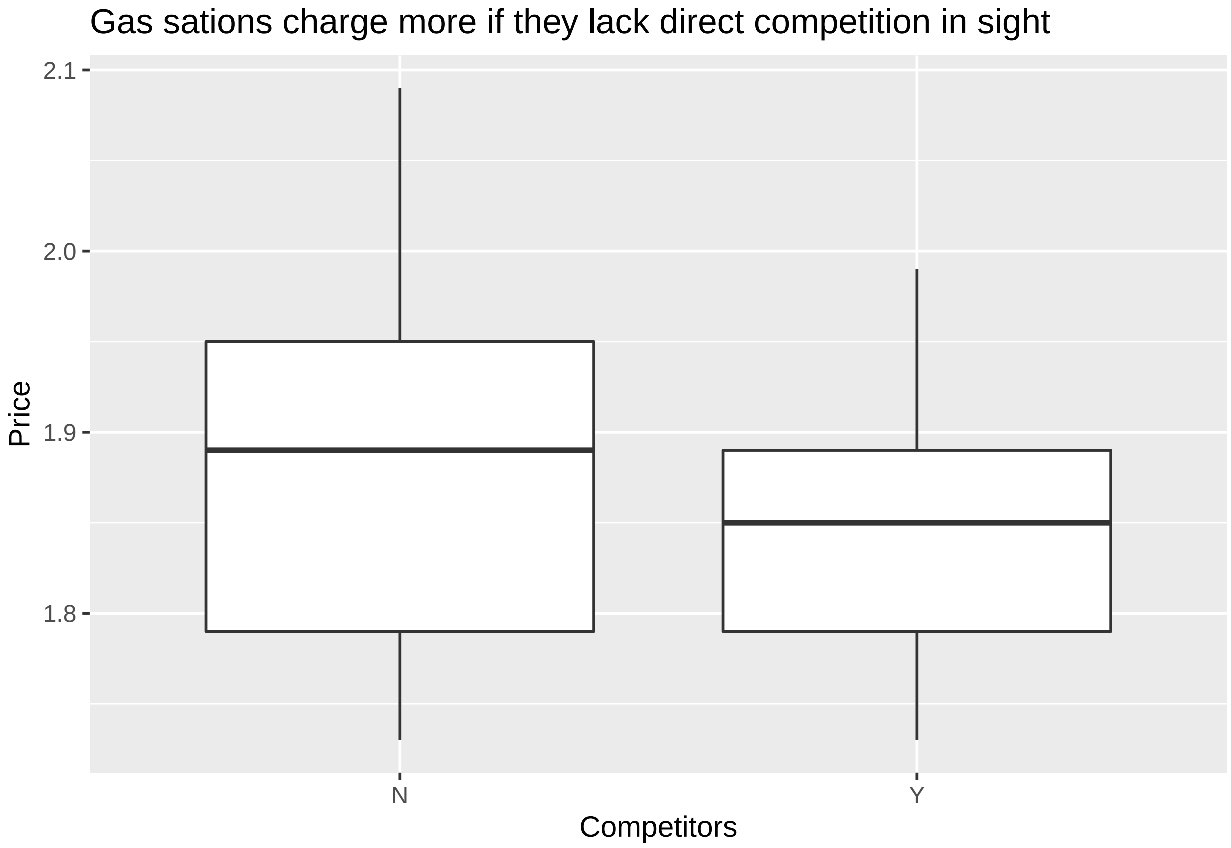
```
GasPrices <- read.csv('GasPrices.csv')
sclass <- read.csv('sclass.csv')
```

Question 1) Data visualization: gas prices

This problem is about making simple plots and telling stories using those plots using the GasPrices.csv data set from the class website. This data set contains data from 101 gas stations in the Austin area in 2016 and we will be using it to determine which theories seem plausible.

Gas sations charge more if they lack direct competition in sight

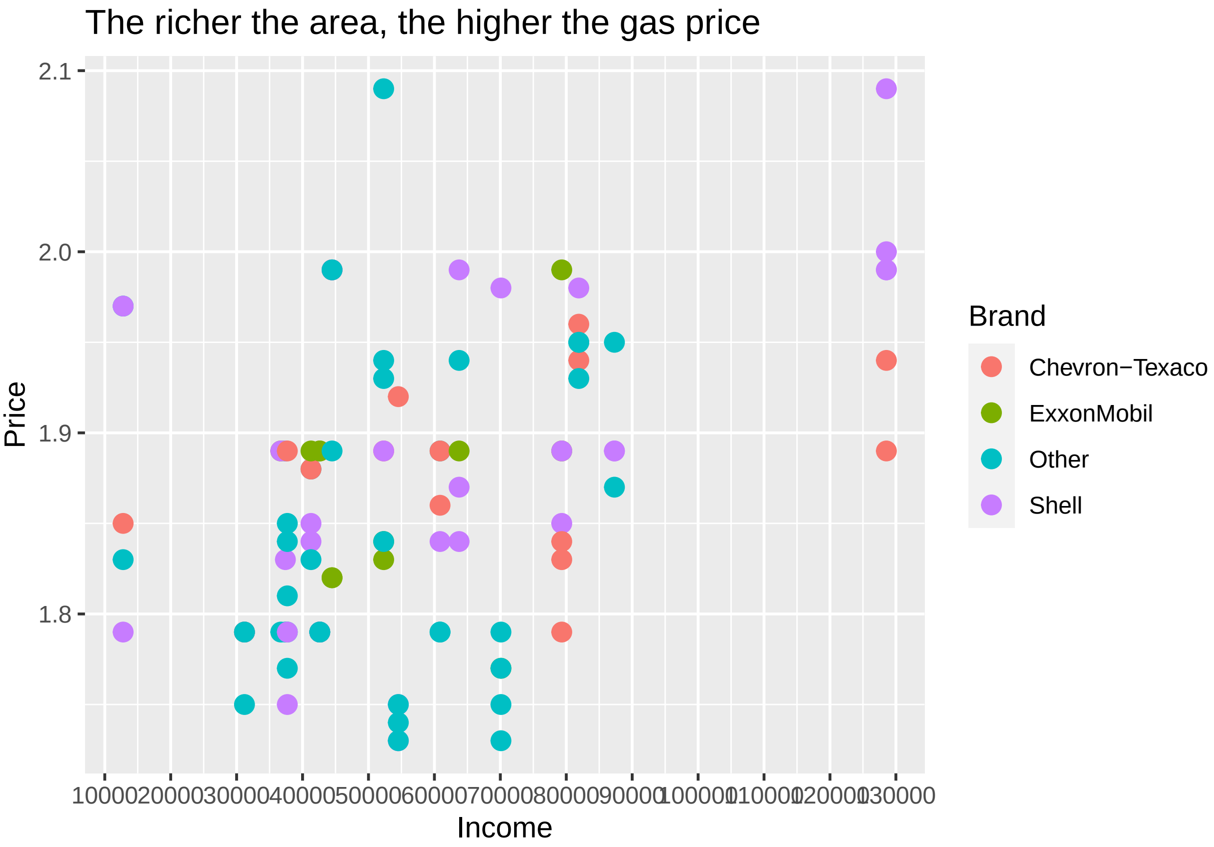
```
CompPrices = GasPrices %>%
  group_by(Competitors) %>%
  summarize(Price)

ggplot(data = CompPrices)+
  geom_boxplot(mapping = aes(x=Competitors, y=Price)) +
  labs(title = 'Gas sations charge more if they lack direct competition in sight')
```



Based the barplot, we can see that there is indeed evidence supporting the claim that the lack of direct competition leads to gas stations charging more. The black bar in the middle of box plots represents the median price, which is higher for the gas stations that do not have competitors in sight, around $1.88 per gallon, compared to the gas stations who have some competition, around $ 1.85 per gallon. Furthermore, the interquantile range for the gas stations lacking competition is greater than that of those with competition, suggesting that there is a greater variation in prices for the competitive gas stations. What is interesting to note that both competing and non-competing gas stations have the same bottom quartile value.

The richer the area, the higher the gas price (scatter plot).

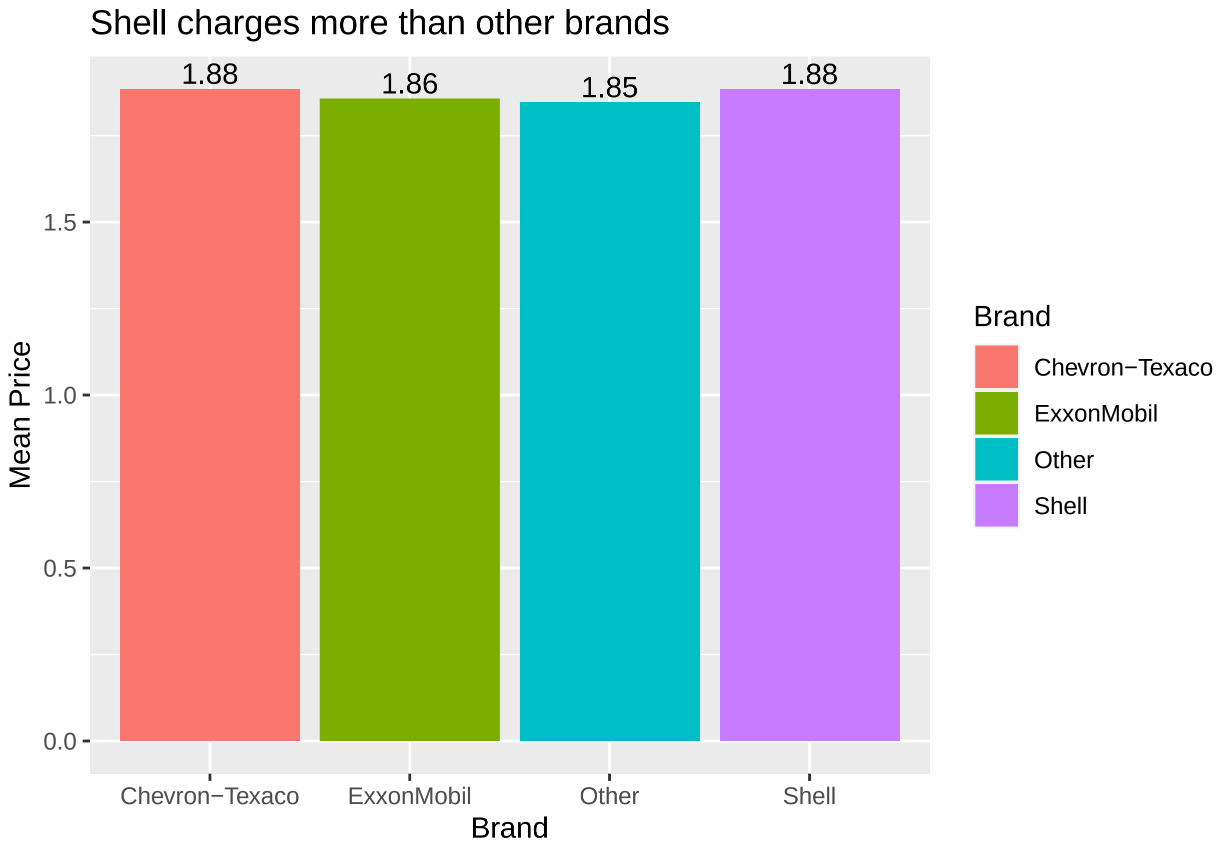
```
ggplot(data = GasPrices)+
  geom_point(mapping = aes(x = Income, y = Price, color = Brand),size =3) +
  scale_x_continuous(breaks=seq(0,130000, 10000))+
labs(title = 'The richer the area, the higher the gas price')
```



The above scatterplot does seems to suggest an positive correlation between the median household income of of where the gas station is located and the price of gasoline. However, the scatterplot also seems to suggest that there is an increase in the price of gasoline in lower income areas as can be seen at the $10 000 - $15 000 Median income range. This could be due to two things. The first is that these might simply be outliners in the dataset. The second, and the most probable reason, is that these gas stations have higher prices because the income elasticity for the citizens that live in these areas are significantly lower. This could be due to the fact that these citizens are more likely to wait until the Empty Fuel Tank signal turns on before getting gas due to budgetary constraints, and thus preventing them from traveling to a cheaper gas station. This is simply a conjecture and would need to be studied further.

Shell charges more than other brands (bar plot).

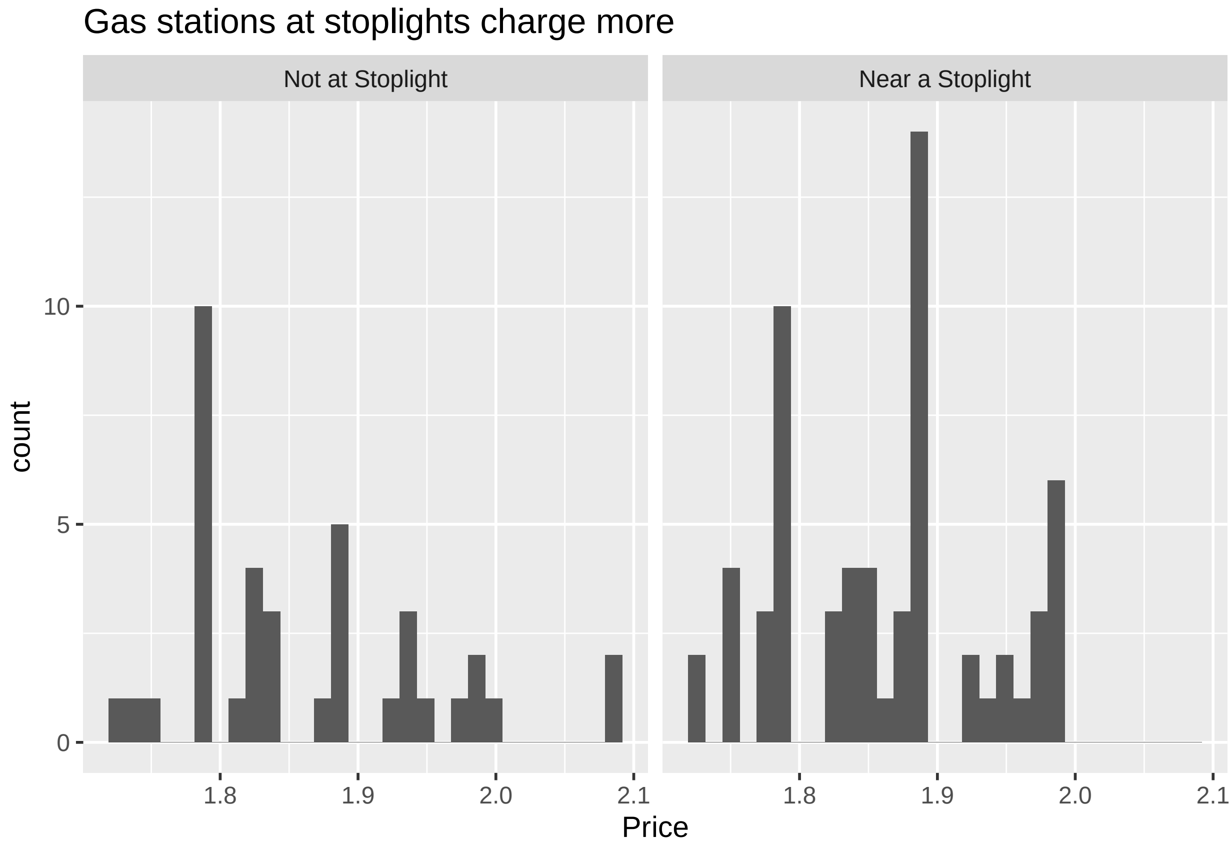
```
Price_Brand = GasPrices %>%
  group_by(Brand) %>%
  summarize(mean_price=mean(Price))


Price_BrandG <- ggplot(data = Price_Brand, aes(x=Brand, y=mean_price, fill=Brand))+
  geom_bar(stat = "identity") +
  ylab('Mean Price') +
  geom_text(aes(label=round(mean_price,digits = 2)), position=position_dodge(width=0.9), vjust=-0.25)
Price_BrandG+
labs(title = 'Shell charges more than other brands')
```



It would seems like Shell does not charge more than other brands, when we compare the average gas price between the various brands as we can see that the average price of gasoline for Chevron-Texaco is the same as Shell.

Gas stations at stoplights charge more (faceted histogram).

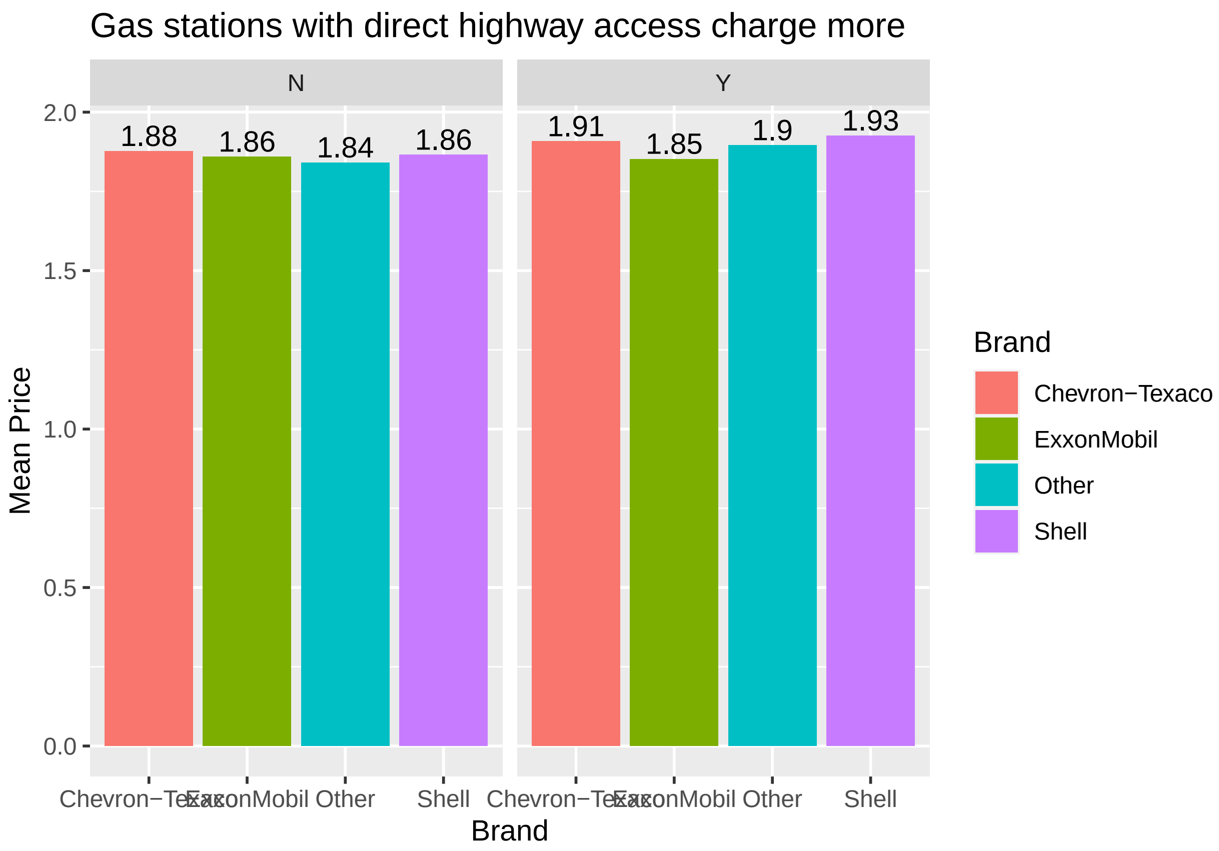
```
labels <- c(N= 'Not at Stoplight', Y = 'Near a Stoplight')
test1 <- ggplot(data = GasPrices)+
  geom_histogram(aes(x=Price), bins = 30) +
  facet_wrap(~ Stoplight, labeller = labeller(Stoplight = labels))
test1 +
labs(title = 'Gas stations at stoplights charge more')
```



The evidence does not seem to support the claim that gas stations at stoplights charge more than those that are not near a gas station. Although it might seem at first glance to support the statement, we can see that the price of gas not near a stop light range from around $1.75 to $2.09, whereas the gas price for stations near the light range from $1.72 to $1.98. So it would seem that the histogram is providing evidence that gas stations not situated at stoplight charge more, not the other way around as the statement claims. However, this is still wrong because we need to look at the count of gas stations at the particular prices. Although the price of gas at non stoplight gas stations has a greater range, there is not a real concentration of prices for the gas stations, compared to the histogram of the gas stations near a stoplight. Taking this into account, it is feasible that the average price of gas between stations near and away from stoplights might almost be identical.

Gas stations with direct highway access charge more (your choice of plot).

```
Price_Brand = GasPrices %>%
  group_by(Brand, Highway) %>%
  summarize(mean_price=mean(Price))


Price_BrandH <- ggplot(data = Price_Brand, aes(x=Brand, y=mean_price, fill=Brand))+
  geom_bar(stat = "identity") +
  ylab('Mean Price') +
  geom_text(aes(label=round(mean_price,digits = 2)), position=position_dodge(width=0.9), vjust=-0.25)+
  facet_wrap(~ Highway)
Price_BrandH +
labs(title = 'Gas stations with direct highway access charge more')
```



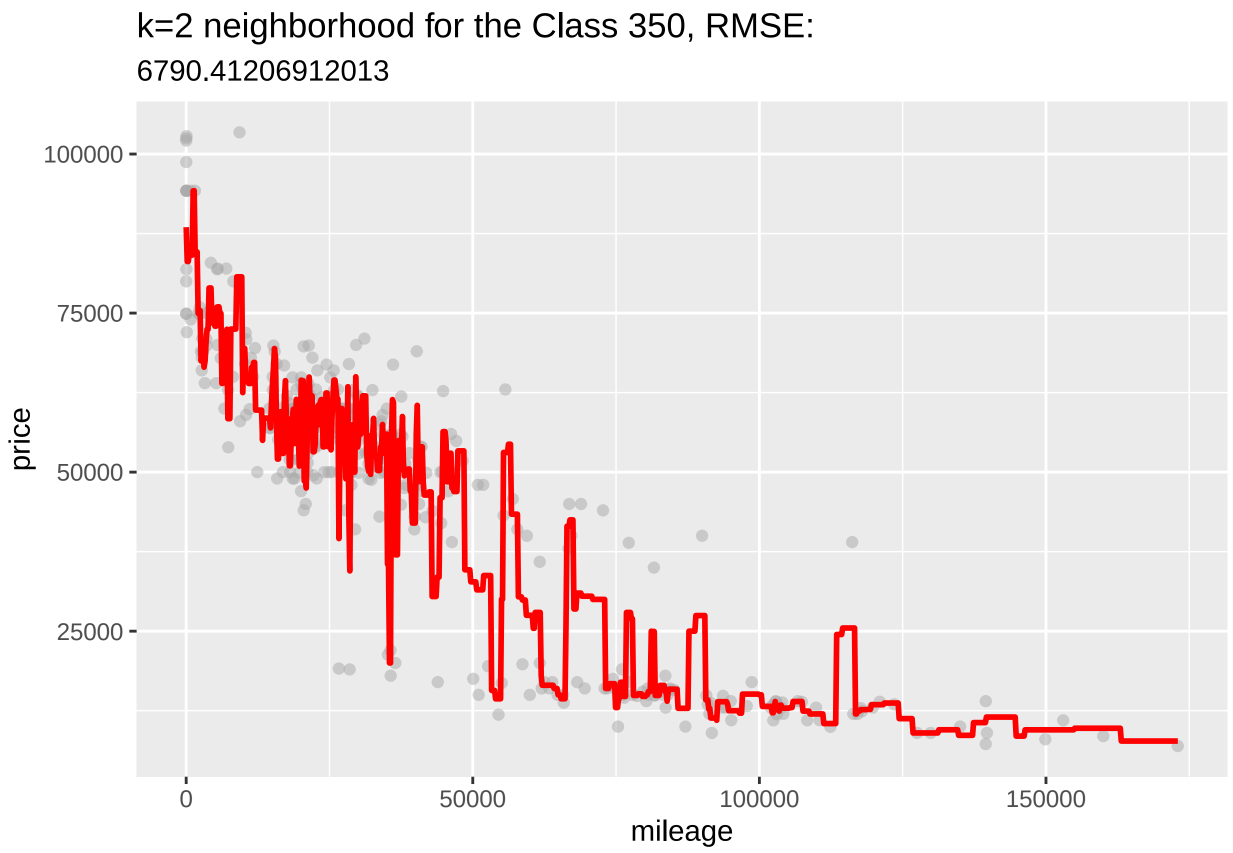
The evidence supports the claim that gas stations with direct highway access charge more, with the exception of ExxonMobil. As we can see from the graphs, Chevron,the other gas stations and Shell have increased their average prices by $0.03, $0.05 and $0.07 respectively. It would be interesting to see if Exxonmobil has some sort of franchise policy that prevents owners from randomly increasing the price, or if these gas stations are situated in areas with higher competition or lower land prices.

Question 4) K-nearest neighbors

For this exercise, we are using the sclass.csv dataset containing data on over 29,000 Mercedes S Class vehicles. For the purpose of this exercise, we are only focusing on two trims, similar to a sub-model designation, the 350 and the 65 AMG.

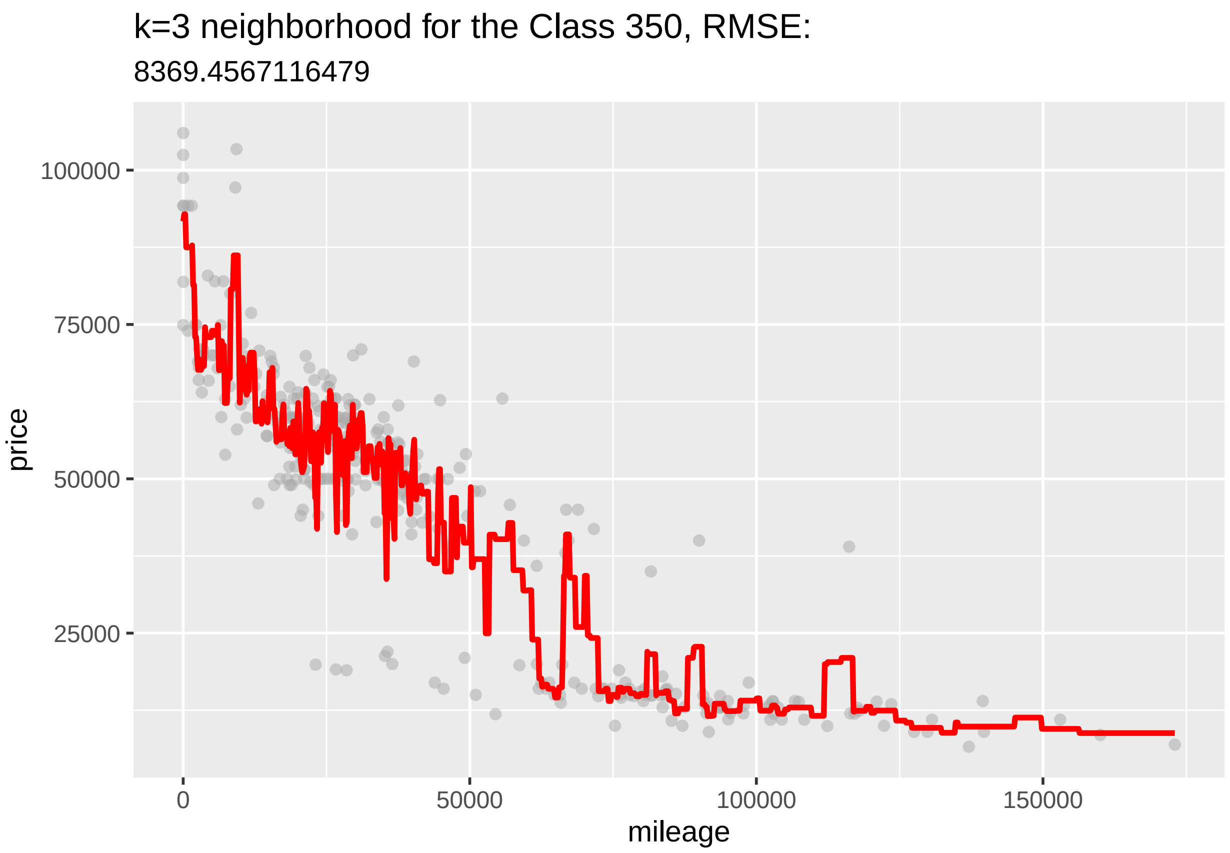
```
Class350 <- sclass %>%
  filter(trim==350)

K= 2
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))
```

```
D_all = Class350; D_all$set = ’test’; D_all$set[train_ind] = ’train’
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = ’k = K’, sub = ’RMSE=RMSE’ ) +
  geom_point(mapping = aes(x = mileage, y = price), color=’darkgrey’, alpha=0.5)
g0 + stat_function(fun=knn_pred, color=’red’, size=1, n=1001) +
  labs(title = ’k=2 neighborhood for the Class 350, RMSE:’, subtitle = RMSE)
```



Above is the scatterplot representing the selected variables from the training set and is overlayed with the predicted curve for the Class 350 at k=2.

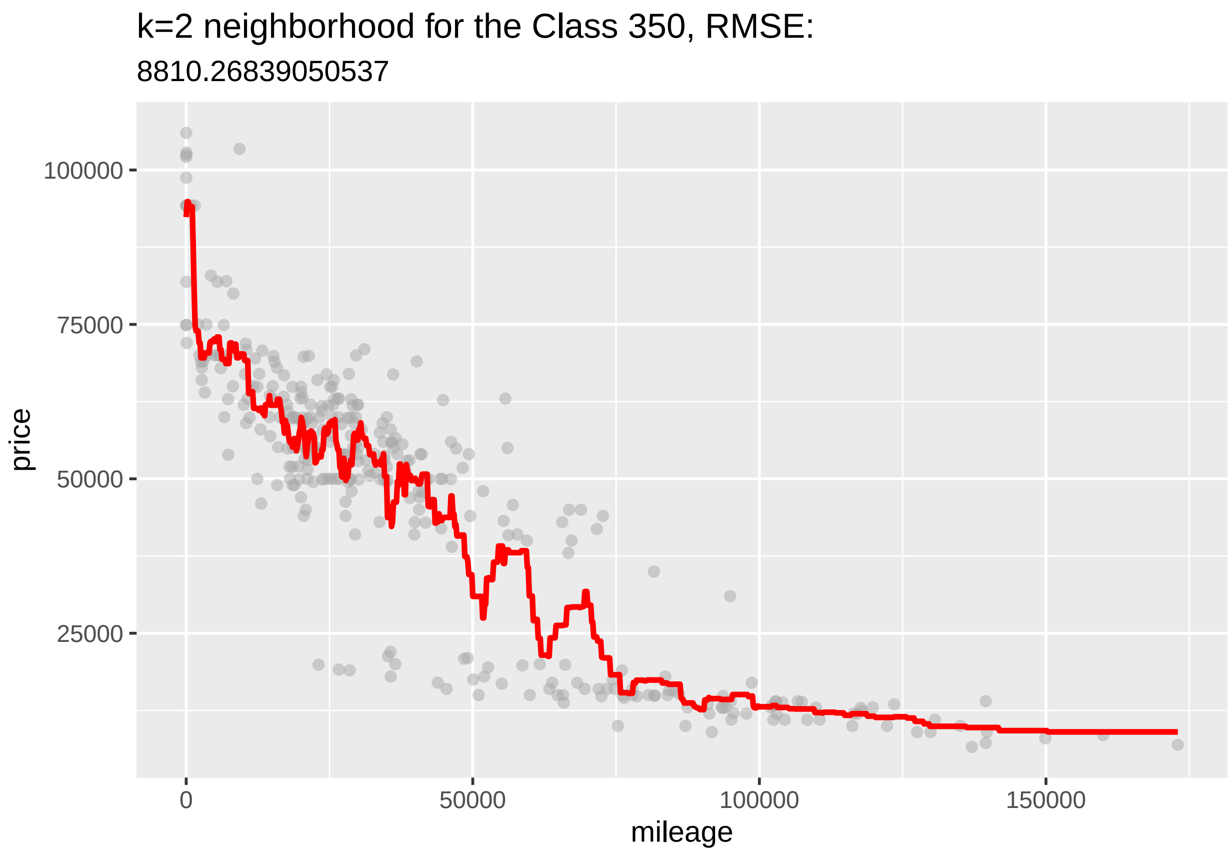
```
K= 3
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = ’test’; D_all$set[train_ind] = ’train’
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = ’k = K’, sub = ’RMSE=RMSE’ ) +
  geom_point(mapping = aes(x = mileage, y = price), color=’darkgrey’, alpha=0.5)
g0 + stat_function(fun=knn_pred, color=’red’, size=1, n=1001) +
  labs(title = ’k=3 neighborhood for the Class 350, RMSE:’, subtitle = RMSE)
```



Above is the scatterplot representing the selected variables from the training set and is overlayed with the predicted curve for the Class 350 at k=3. As we can see the RMSE has increased and the predicted model is becoming smoother.

```
K= 10
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)
```

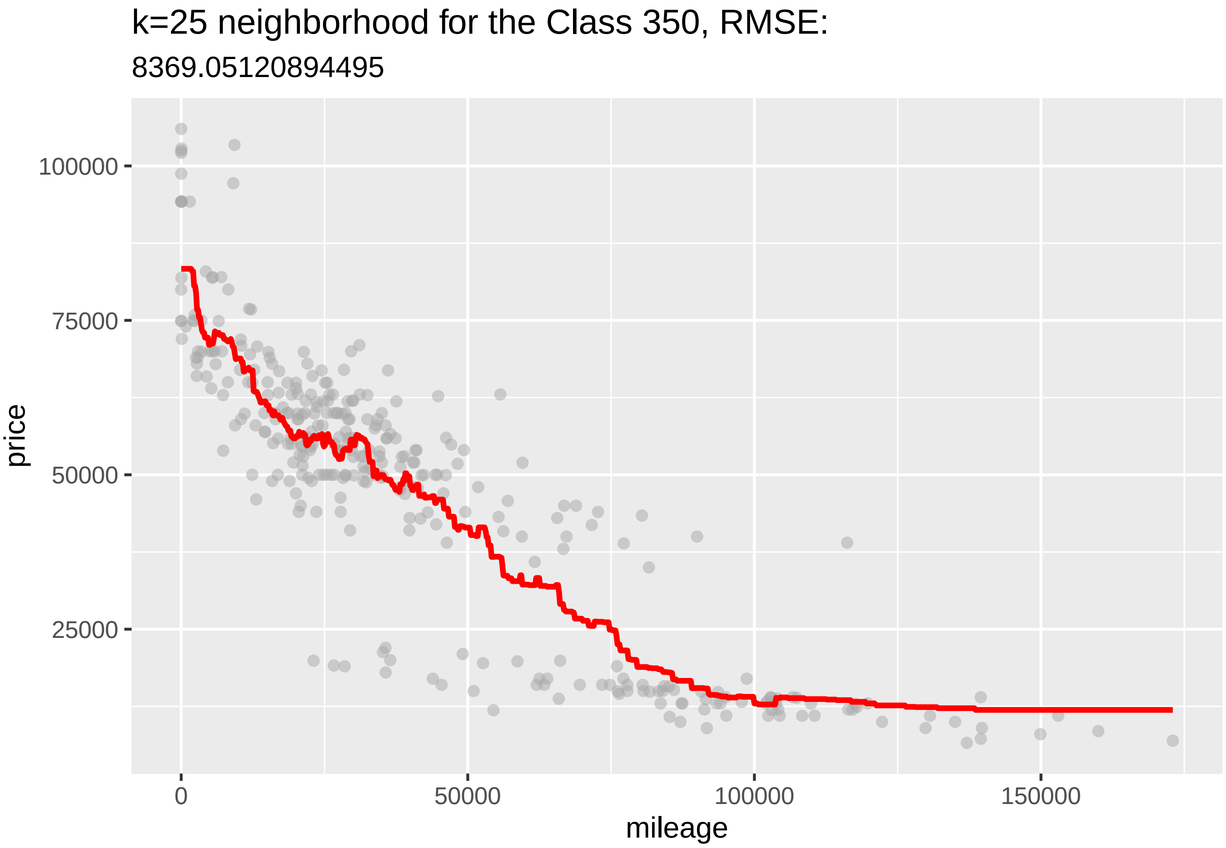
```
g0 = ggplot(data = D_train, title = 'k = K', sub = 'RMSE=RMSE' ) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=2 neighborhood for the Class 350, RMSE:', subtitle = RMSE)
```



Above is the scatterplot representing the selected variables from the training set and is overlayed with the predicted curve for the Class 350 at k=10. The RMSE has significantly increased but we get a better fit

```
K= 25
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}
```

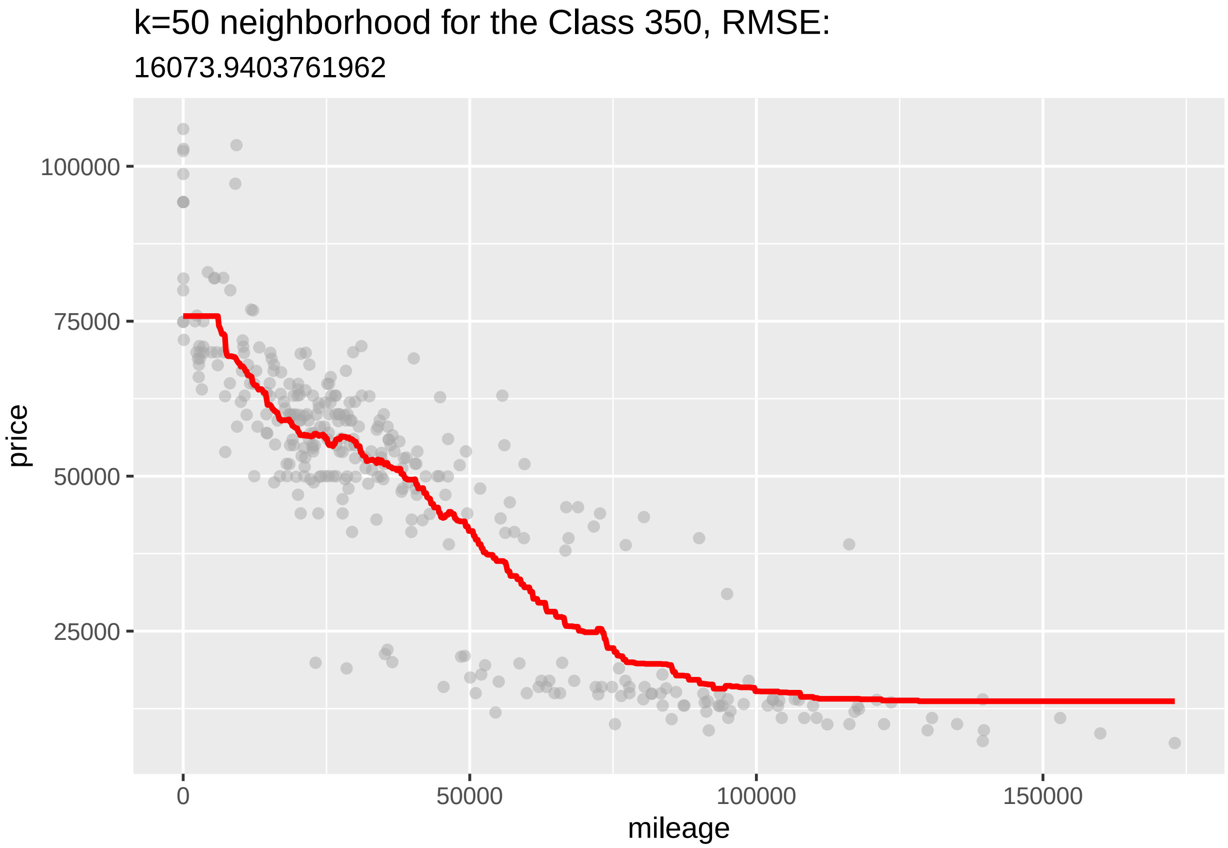
```
knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = 'k = K', sub = 'RMSE=RMSE' ) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=25 neighborhood for the Class 350, RMSE:', subtitle = RMSE)
```



Above is the scatterplot representing the selected variables from the training set and is overlayed with the predicted curve for the Class 350 at k=25. We get a slight increase in the RMSE for a fit that best seems to fit the training set.

```
K= 50
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))
```

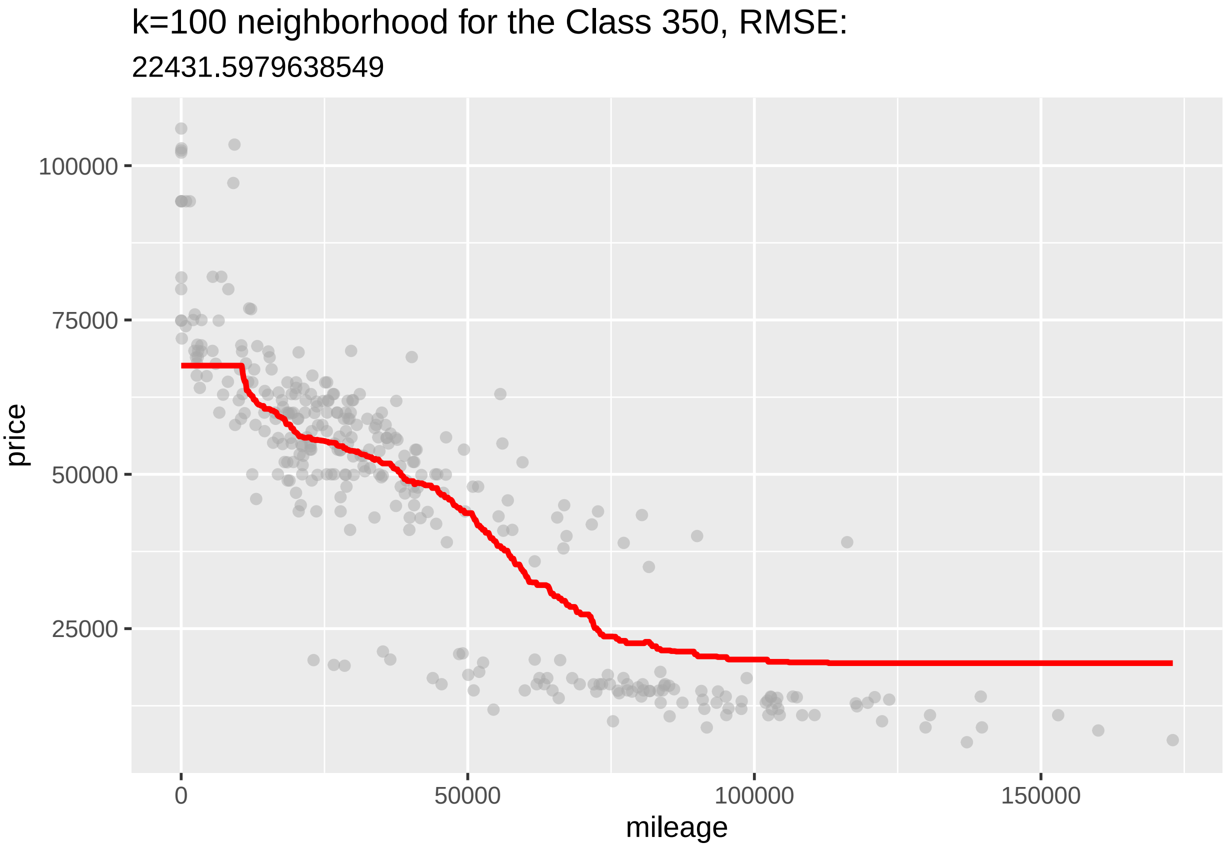
```r
knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = 'k = K', sub = 'RMSE=RMSE' ) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=50 neighborhood for the Class 350, RMSE:', subtitle = RMSE)
```



Above is the scatterplot representing the selected variables from the training set and is overlayed with the predicted curve for the Class 350 at k=50. The RMSE has seen another significant jump, if we look closely at when the mileage is equal to zero, we can see that our predictions are becoming worse near the endpoints of our data. This is consistent with what we know because the KNN takes the average of the 50 nearest data points meaning that the data points situated near the end are going to have the same average.

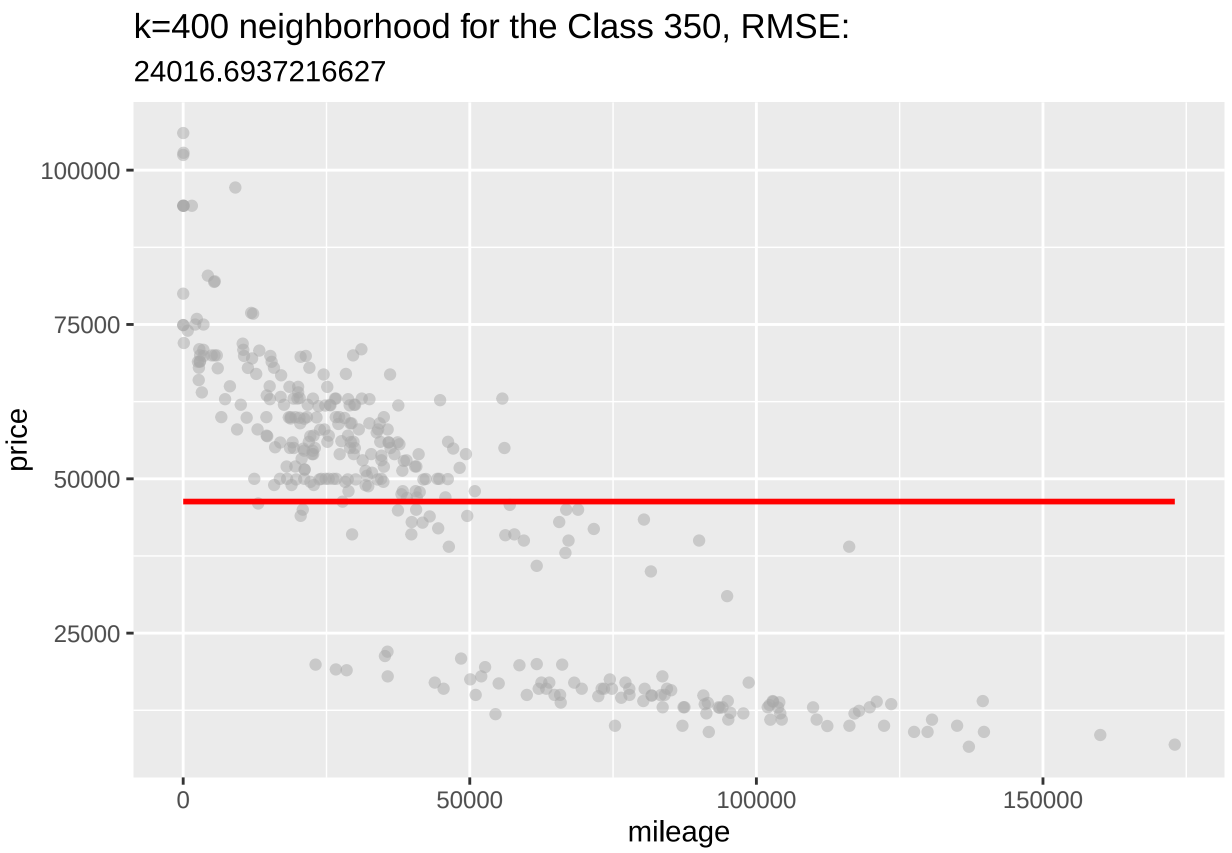
```r
K= 100
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = 'test'; D_all$set[train_ind] = 'train'
```

```
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = ’k = K’, sub = ’RMSE=RMSE’ ) +
  geom_point(mapping = aes(x = mileage, y = price), color=’darkgrey’, alpha=0.5)
g0 + stat_function(fun=knn_pred, color=’red’, size=1, n=1001) +
  labs(title = ’k=100 neighborhood for the Class 350, RMSE:’, subtitle = RMSE)
```



Above is the scatterplot representing the selected variables from the training set and is overlayed with the predicted curve for the Class 350 at k=100. Again the predictions are becoming worse.

```
K= 400
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = ’test’; D_all$set[train_ind] = ’train’
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = ’k = K’, sub = ’RMSE=RMSE’ ) +
  geom_point(mapping = aes(x = mileage, y = price), color=’darkgrey’, alpha=0.5)
g0 + stat_function(fun=knn_pred, color=’red’, size=1, n=1001) +
  labs(title = ’k=400 neighborhood for the Class 350, RMSE:’, subtitle = RMSE)
```



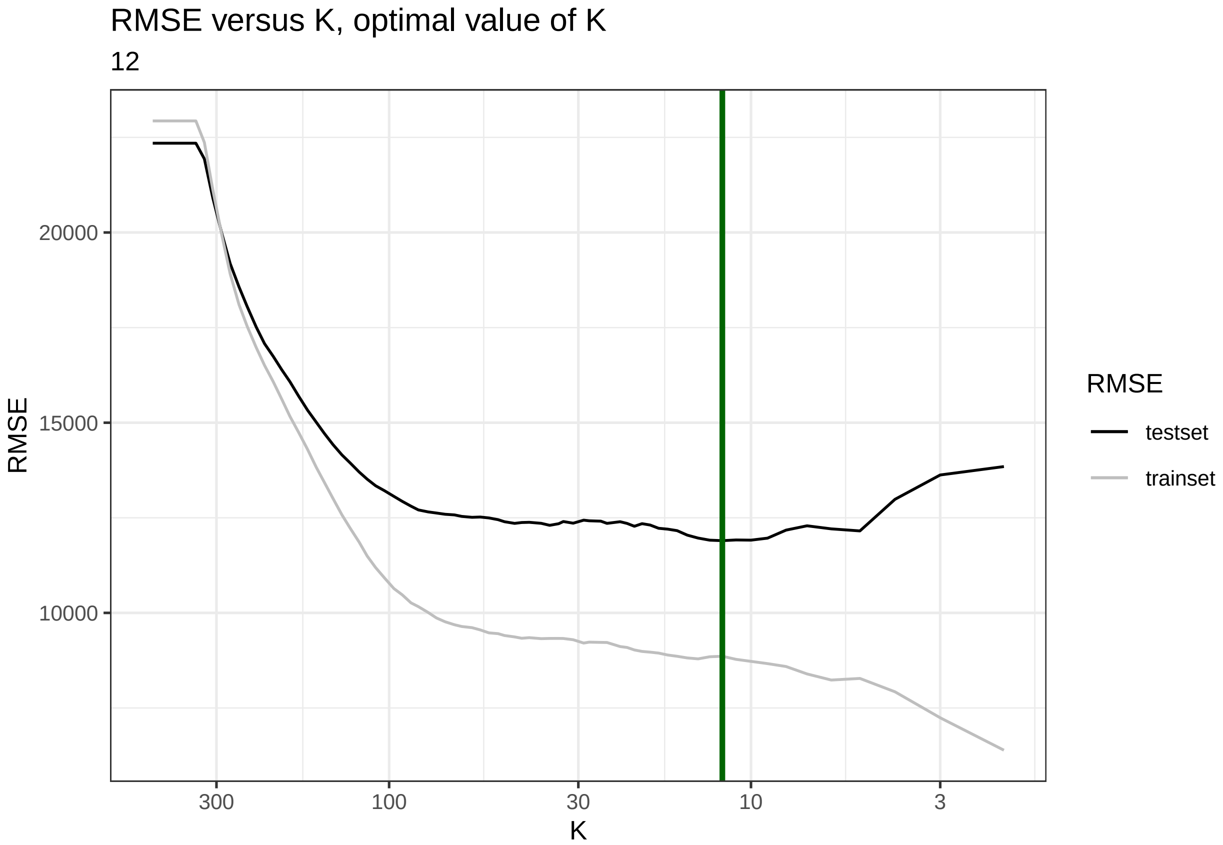
As we have included all the data points with out by selecting 400 neighbors, we are going to have the same prediction no matter the mileage on the car.

In the following graph, I am plotting the RMSE versus K to see where it bottoms out for the Class 350 Trim to find the optimal value of K, which I will then fit to the model.

```
Class350 <- sclass %>%
  filter(trim==350)

N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = ’test’; D_all$set[train_ind] = ’train’
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]


y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))


################RMSE Out-sample

k_grid = unique(round(exp(seq(log(450), log(2), length=100))))
```

```r
rmse_grid_out = foreach(k = k_grid, .combine=’c’) %do% {
  knn_model = knnreg(price ~ mileage, data=D_train, k = k)
  modelr::rmse(knn_model, D_test)
}

rmse_grid_out = data.frame(K = k_grid, RMSE = rmse_grid_out)






###



############RMSE In-sample
k_grid = unique(round(exp(seq(log(450), log(2), length=100))))

rmse_grid_in = foreach(k = k_grid, .combine=’c’) %do% {
  knn_model = knnreg(price ~ mileage, data=D_train, k = k)
  modelr::rmse(knn_model, D_train)
}

revlog_trans <- function(base = exp(1)) {
  require(scales)
  ## Define the desired transformation.
  trans <- function(x){
    -log(x, base)
  }
  ## Define the reverse of the desired transformation
  inv <- function(x){
    base^(-x)
  }
  ## Creates the transformation
  scales::trans_new(paste("revlog-", base, sep = ""),
                    trans,
                    inv,  ## The reverse of the transformation
                    log_breaks(base = base), ## default way to define the scale breaks
                    domain = c(1e-100, Inf)
  )
}

rmse_grid_in = data.frame(K = k_grid, RMSE = rmse_grid_in)

######### Graph both

p_out = ggplot(data=rmse_grid_out) +
  theme_bw(base_size = 10) +
  geom_path(aes(x=K, y=RMSE, color='testset'), size=0.5) +
  scale_x_continuous(trans=revlog_trans(base = 10))

ind_best = which.min(rmse_grid_out$RMSE)
k_best = k_grid[ind_best]
```

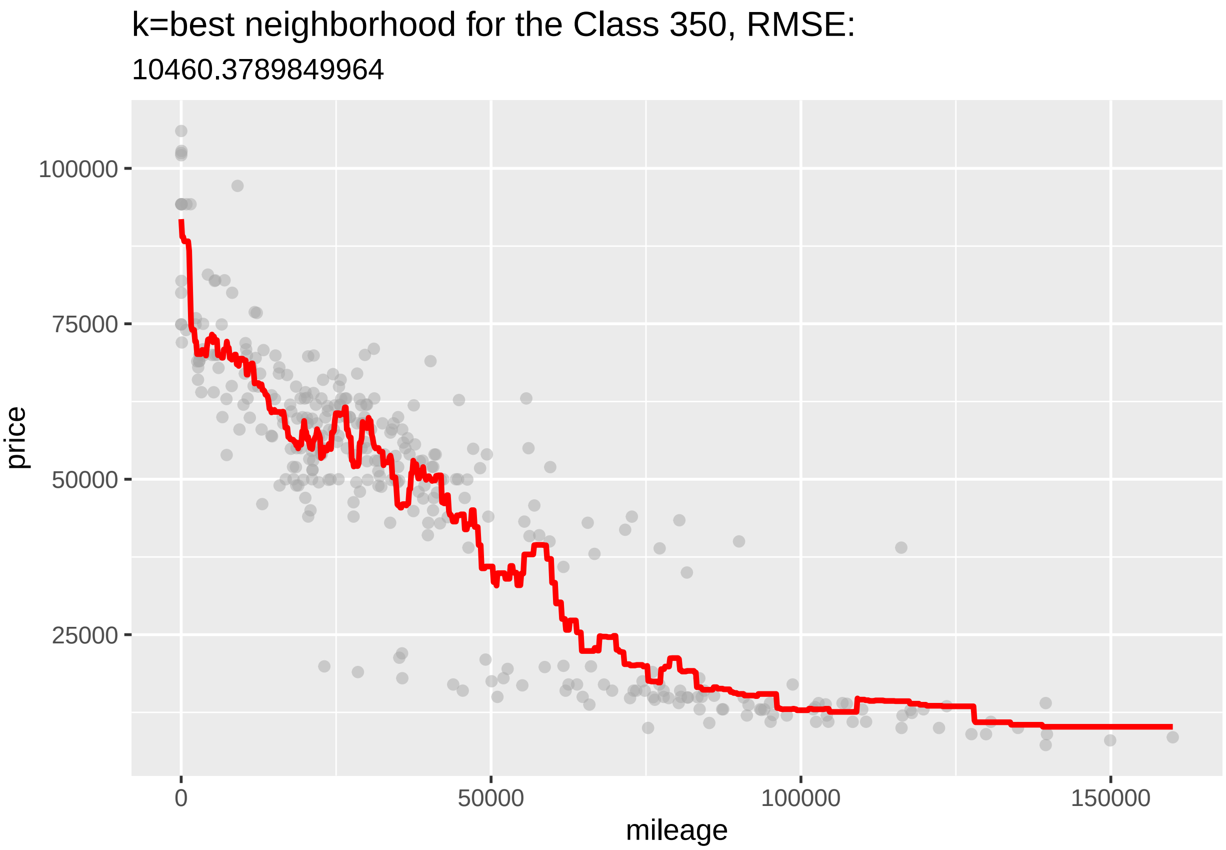
```
p_out + geom_path(data=rmse_grid_in, aes(x=K, y=RMSE, color='trainset'),size=0.5) +
  scale_colour_manual(name="RMSE",
                      values=c(testset="black", trainset="grey")) +
  geom_vline(xintercept=k_best, color='darkgreen', size=1) +
  labs(title = 'RMSE versus K, optimal value of K', subtitle = k_best)
```



As we can see from the graph above, the optimal value of K is 70, this might differ for you if you were to rerun the program or are running for the first time as the K value has varied between 67 and 70 upon multiple reruns. Below is a graph showing the plot of the fitted model.

```
K= k_best
N = nrow(Class350)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class350; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class350[train_ind,]
D_test = Class350[-train_ind,]


y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
```

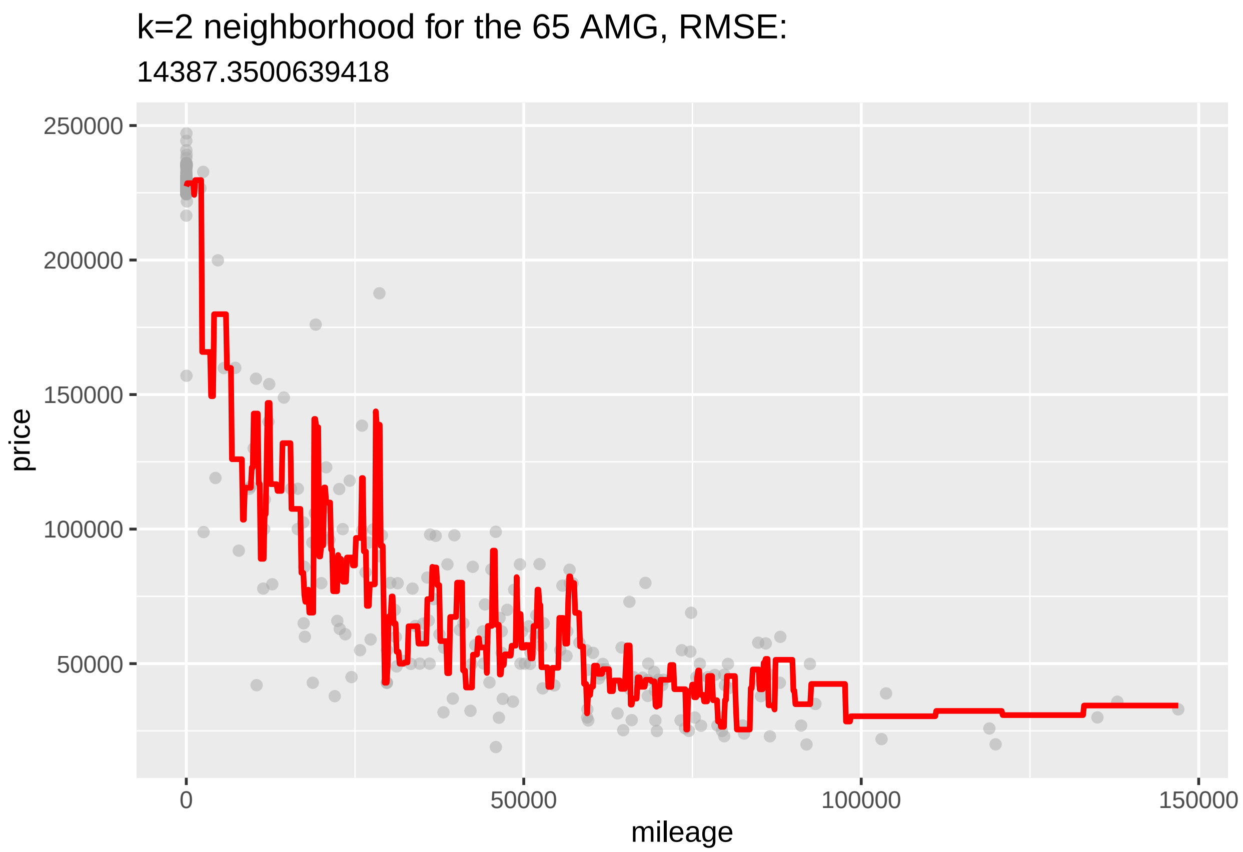
```
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = D_train, title = 'k = K', sub = 'RMSE=RMSE' ) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=best neighborhood for the Class 350, RMSE:', subtitle = RMSE)
```



Similarly for the 65 AMG, we get the following graphs:

```
Class65AMG <- sclass %>%
  filter(trim=='65 AMG')

K= 2
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]
```

```
y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_test, k = K)
RMSE <- modelr::rmse(knn_model, D_test)

g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=2 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

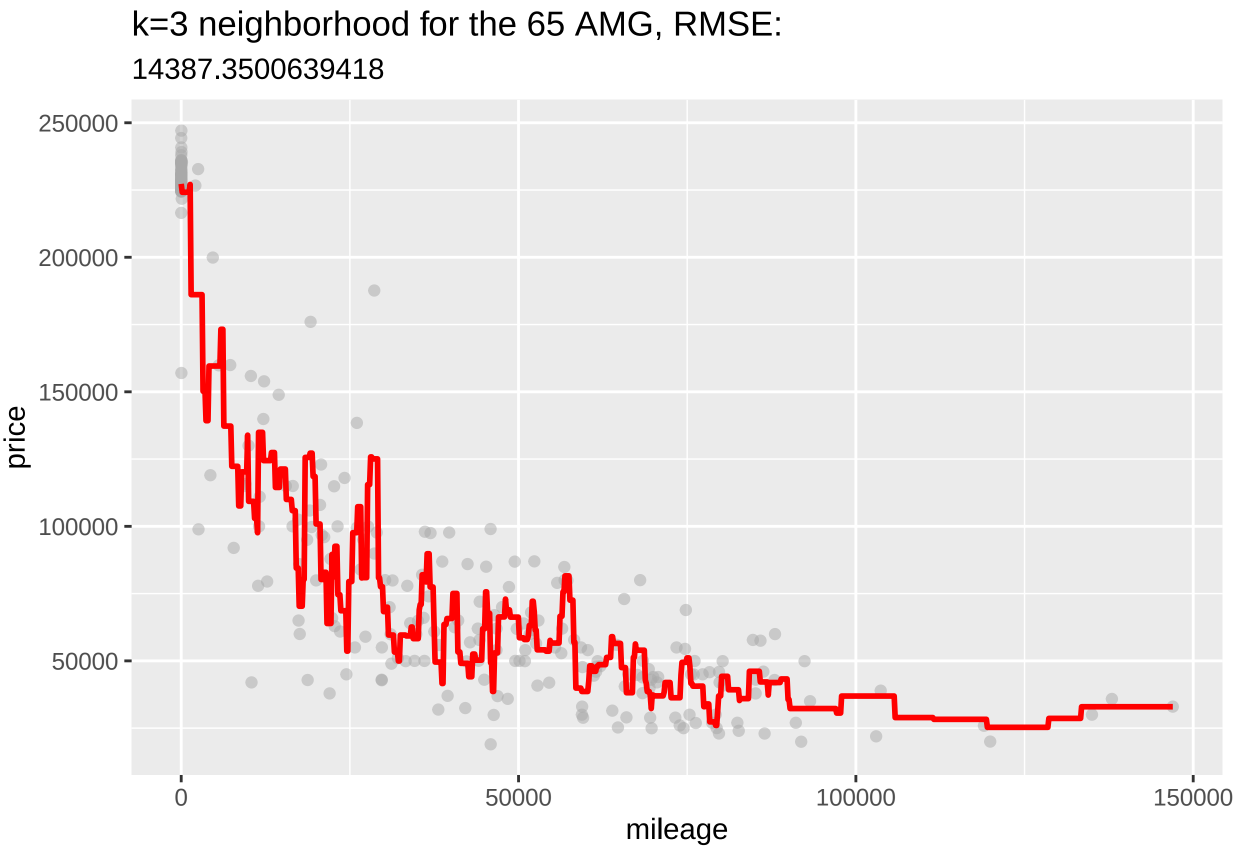


```
K= 3
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
```

```
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 22132.83
```

```
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=3 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

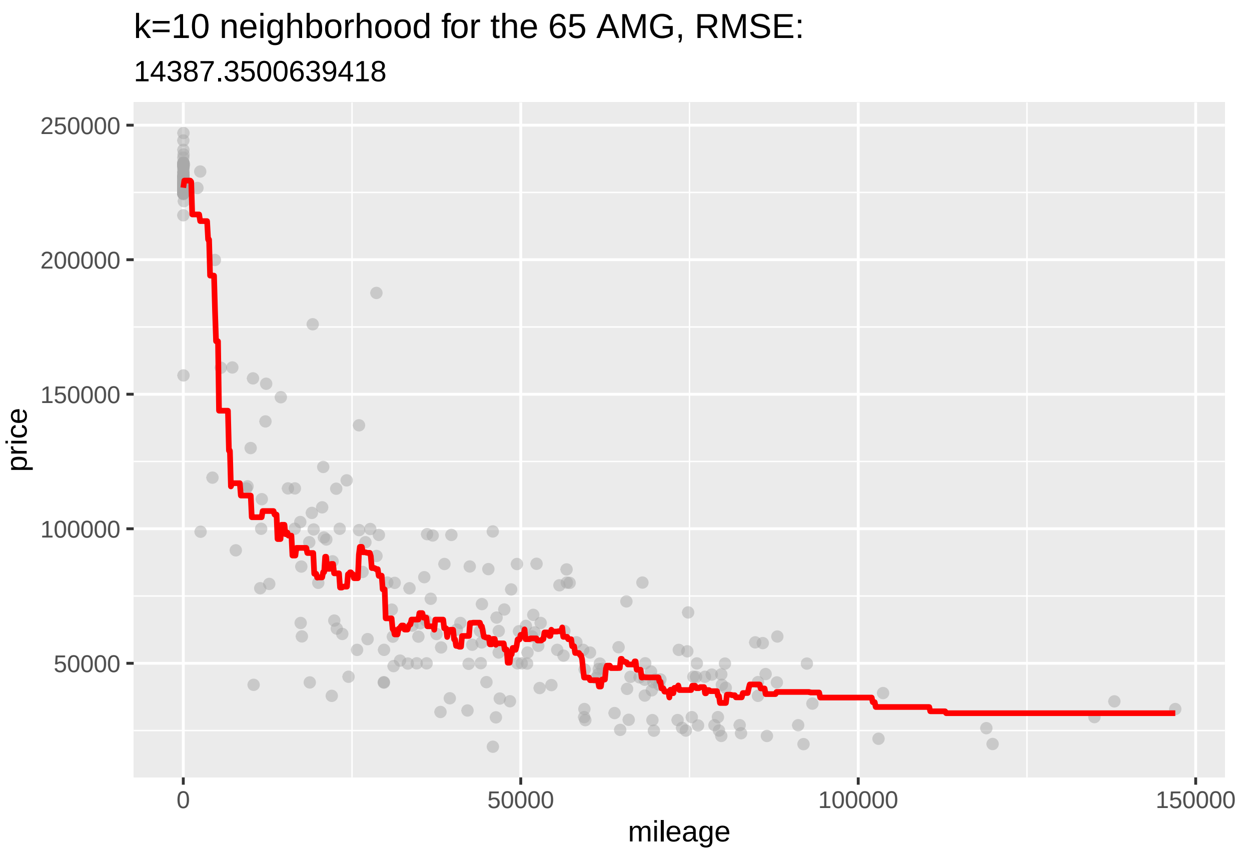


```
K= 10
N = nrow(Class65AMG)
```

```
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 22479.21
```

```
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=10 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

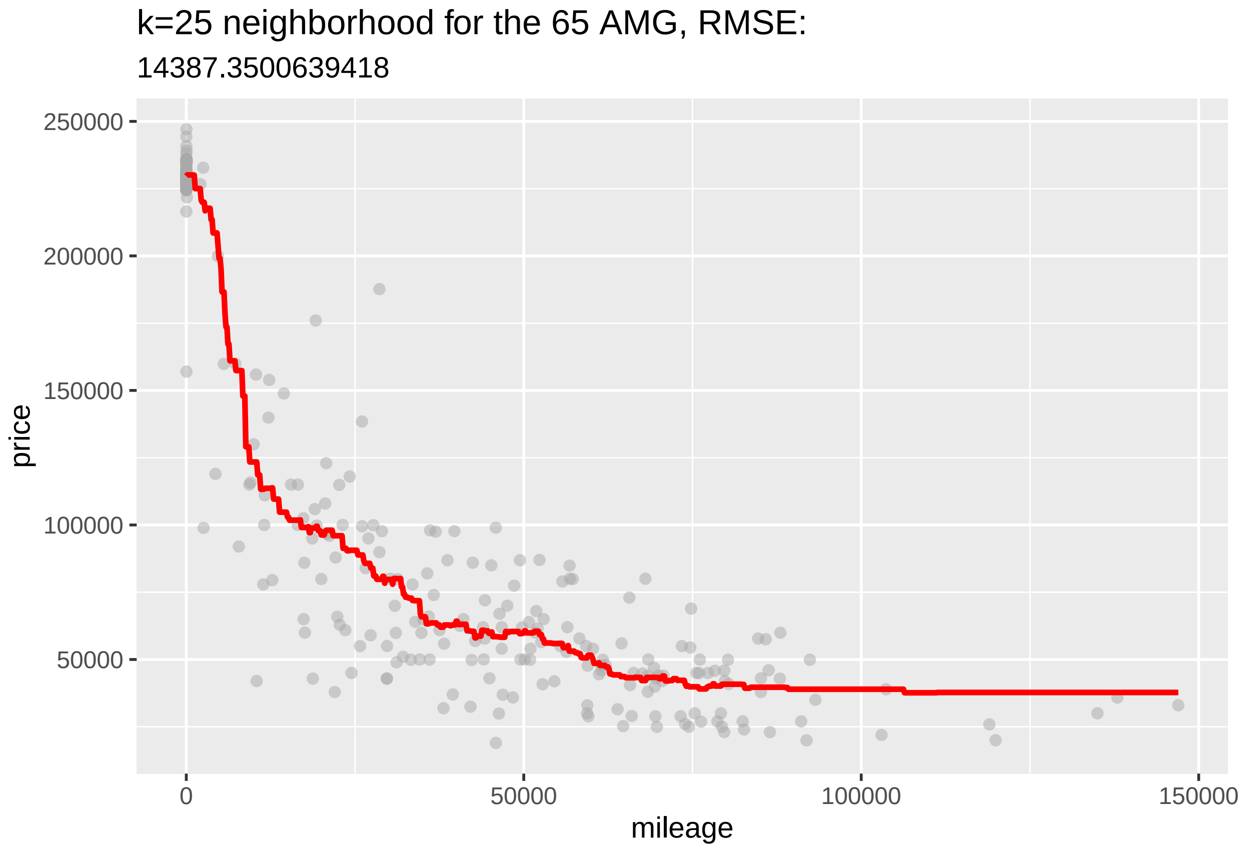


k=10 neighborhood for the 65 AMG, RMSE:
14387.3500639418

```
K= 25
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 19307.29
```

```
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=25 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

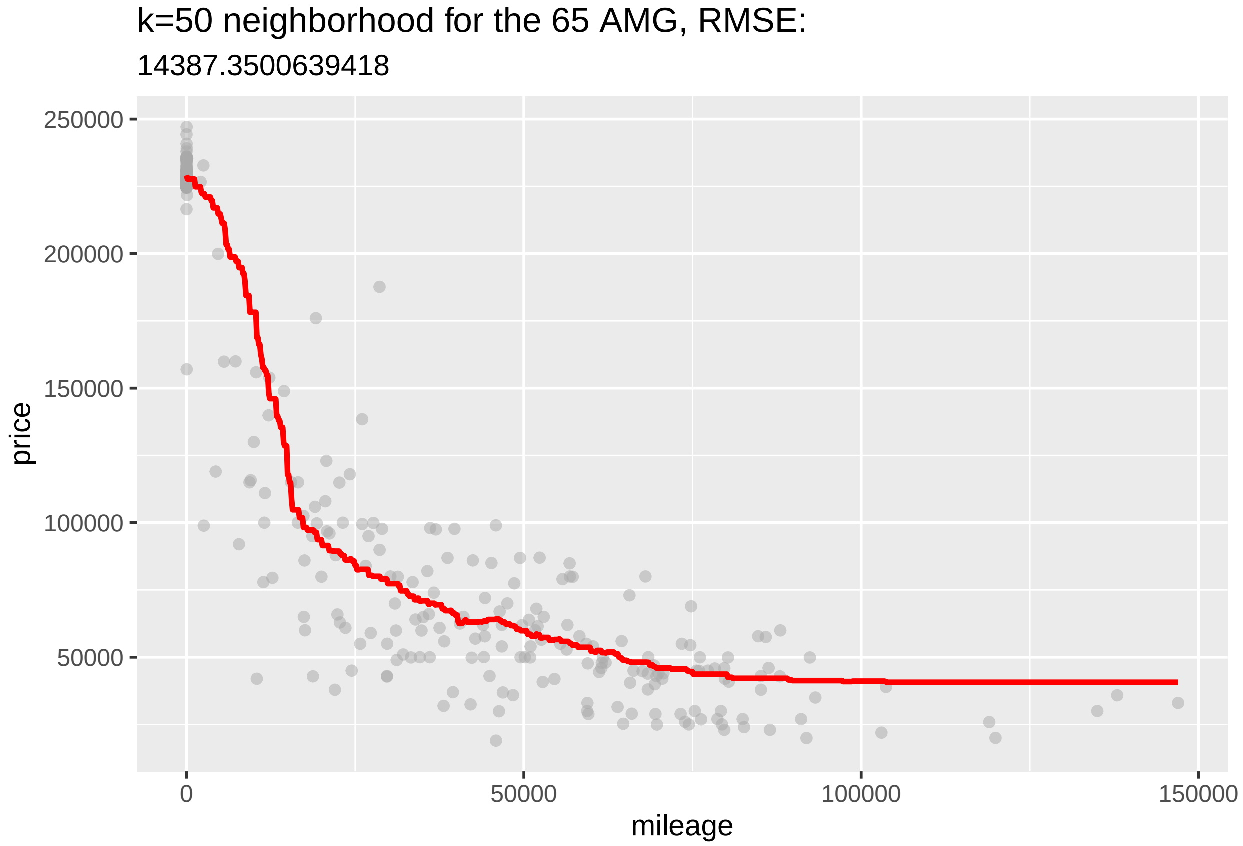


```
K= 50
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
```

```
modelr::rmse(knn_model, D_test)
```

```
## [1] 23782.51
```

```
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=50 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

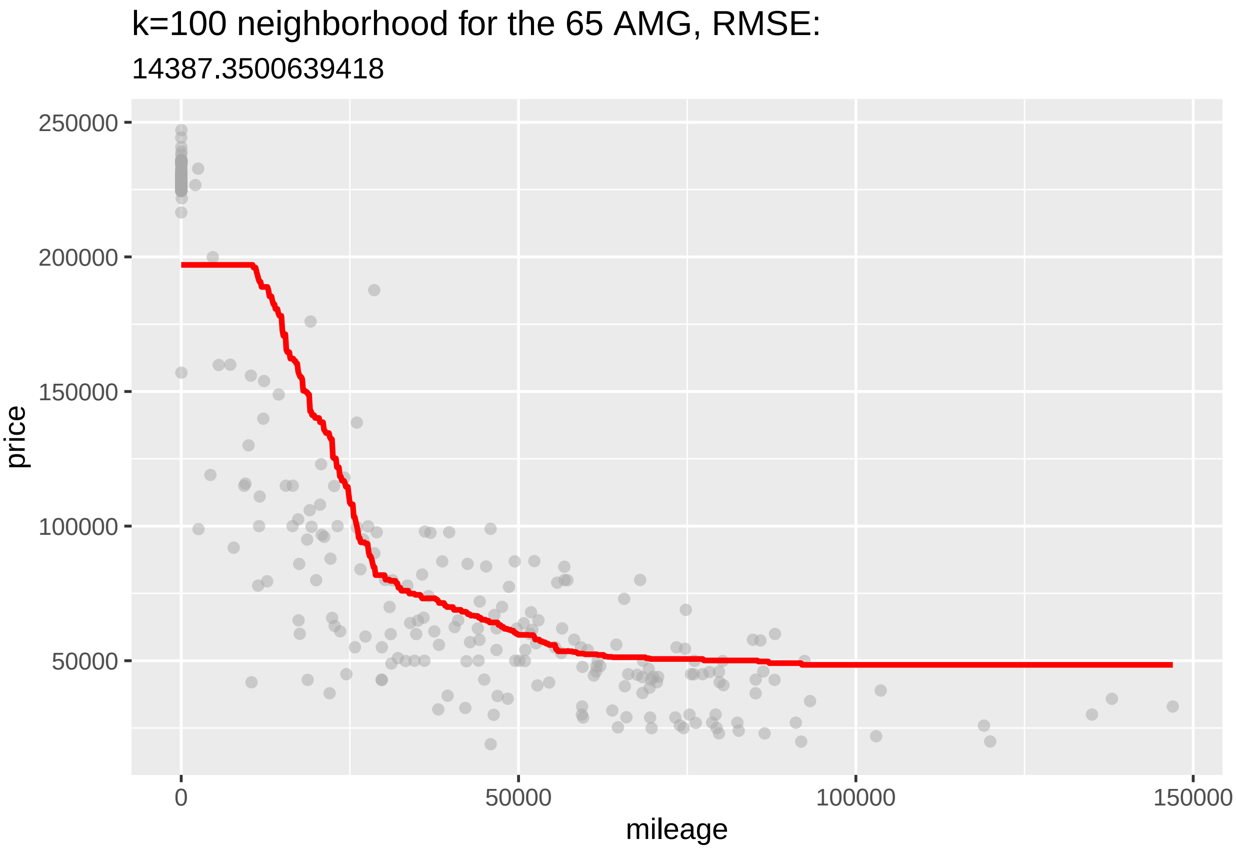


```
K= 100
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
```

```r
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 38239.05
```

```r
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=100 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

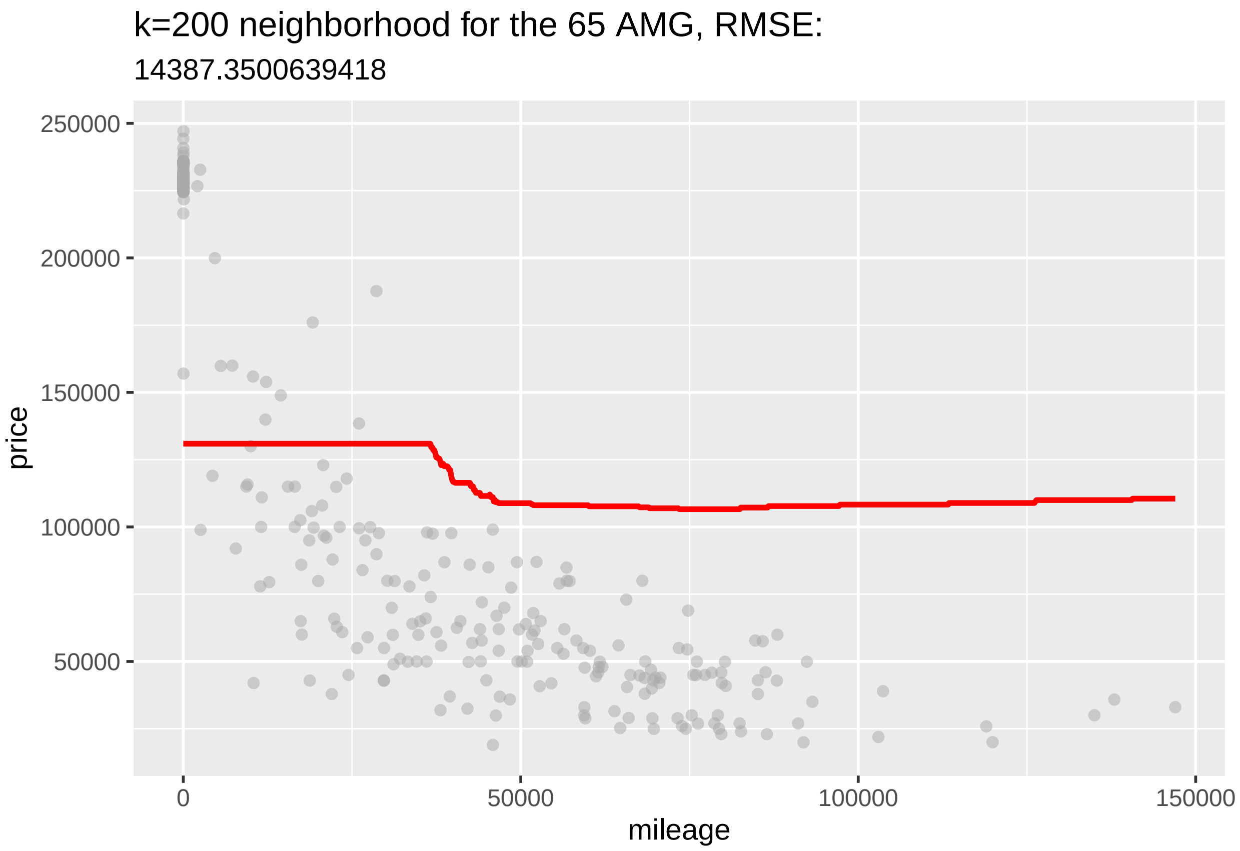


```r
K= 200
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
```

```
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 75355.02
```

```
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=200 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

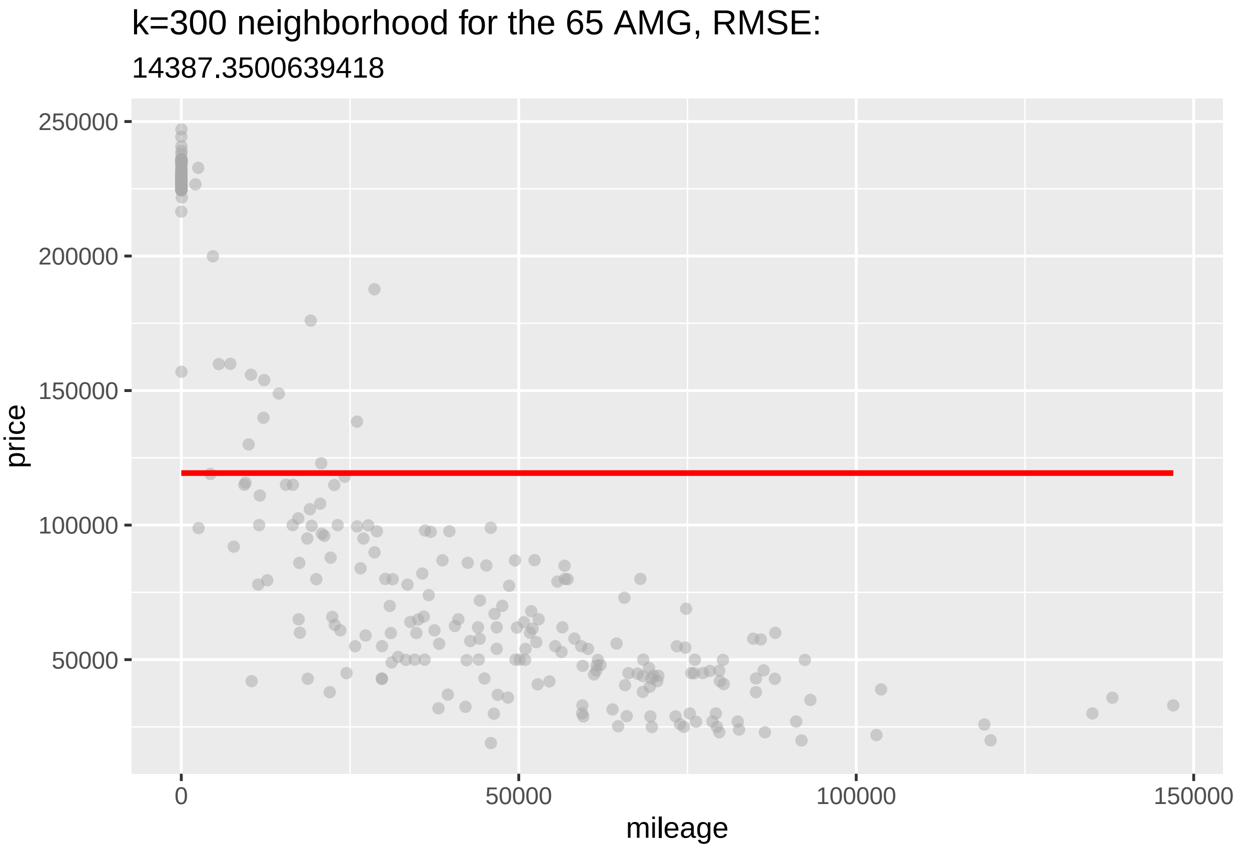


```
K= 300
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))
```

```
D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 77978.24
```

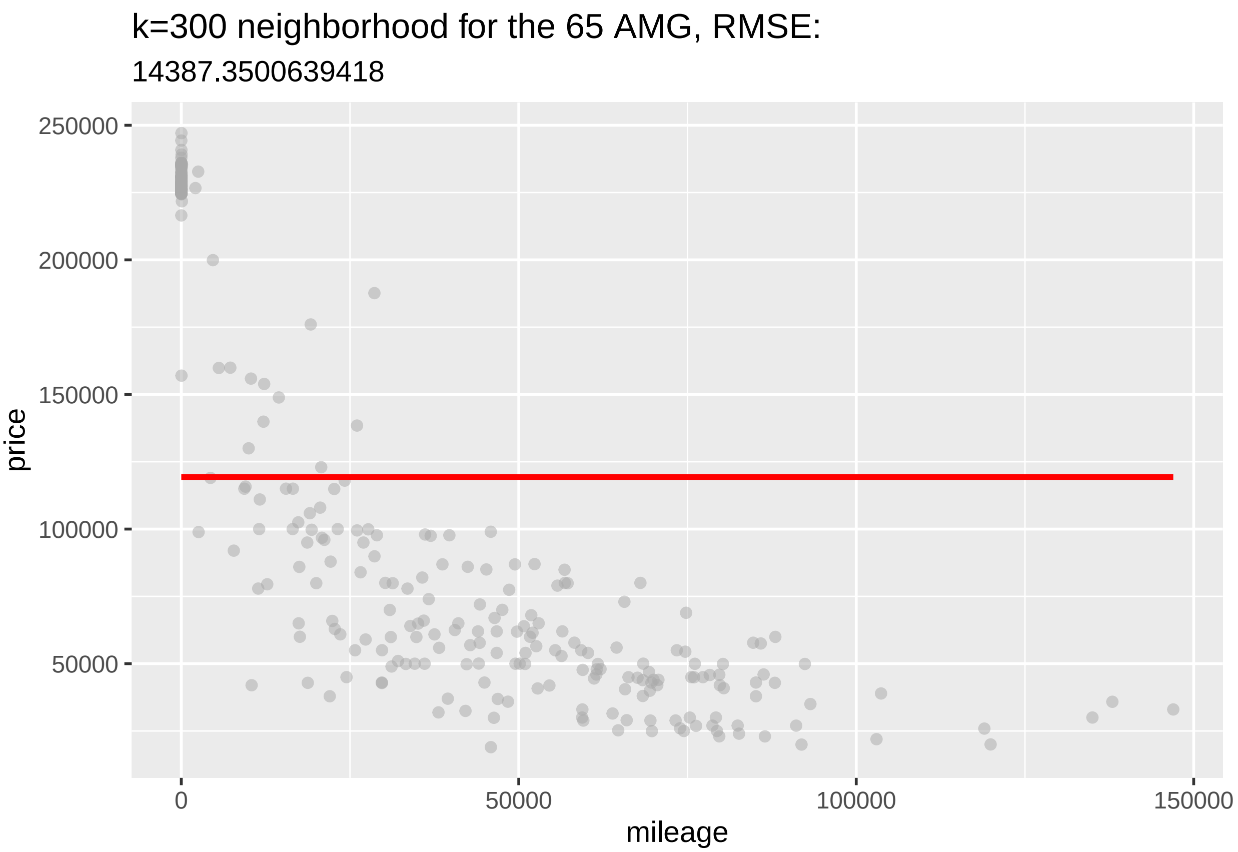
```
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=300 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```



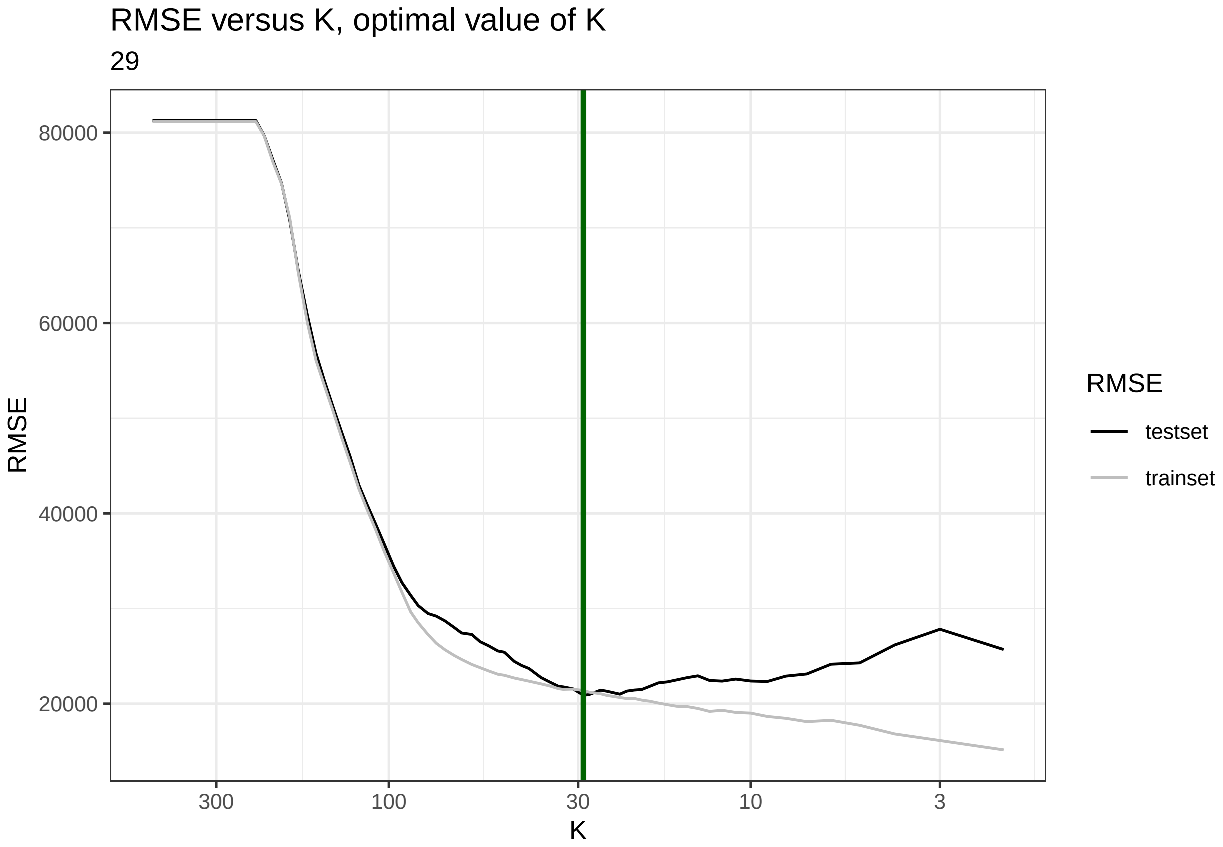
```r
K= 300
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = 'test'; D_all$set[train_ind] = 'train'
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))

knn = knnreg(X_train, y_train, k=K)
knn_pred = function(x) {
  predict(knn, newdata=data.frame(mileage=x))
}

knn_model = knnreg(price ~ mileage, data=D_train, k = K)
modelr::rmse(knn_model, D_test)
```

```
## [1] 75476.7
```

```r
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 + stat_function(fun=knn_pred, color='red', size=1, n=1001) +
  labs(title = 'k=300 neighborhood for the 65 AMG, RMSE:', subtitle = RMSE)
```

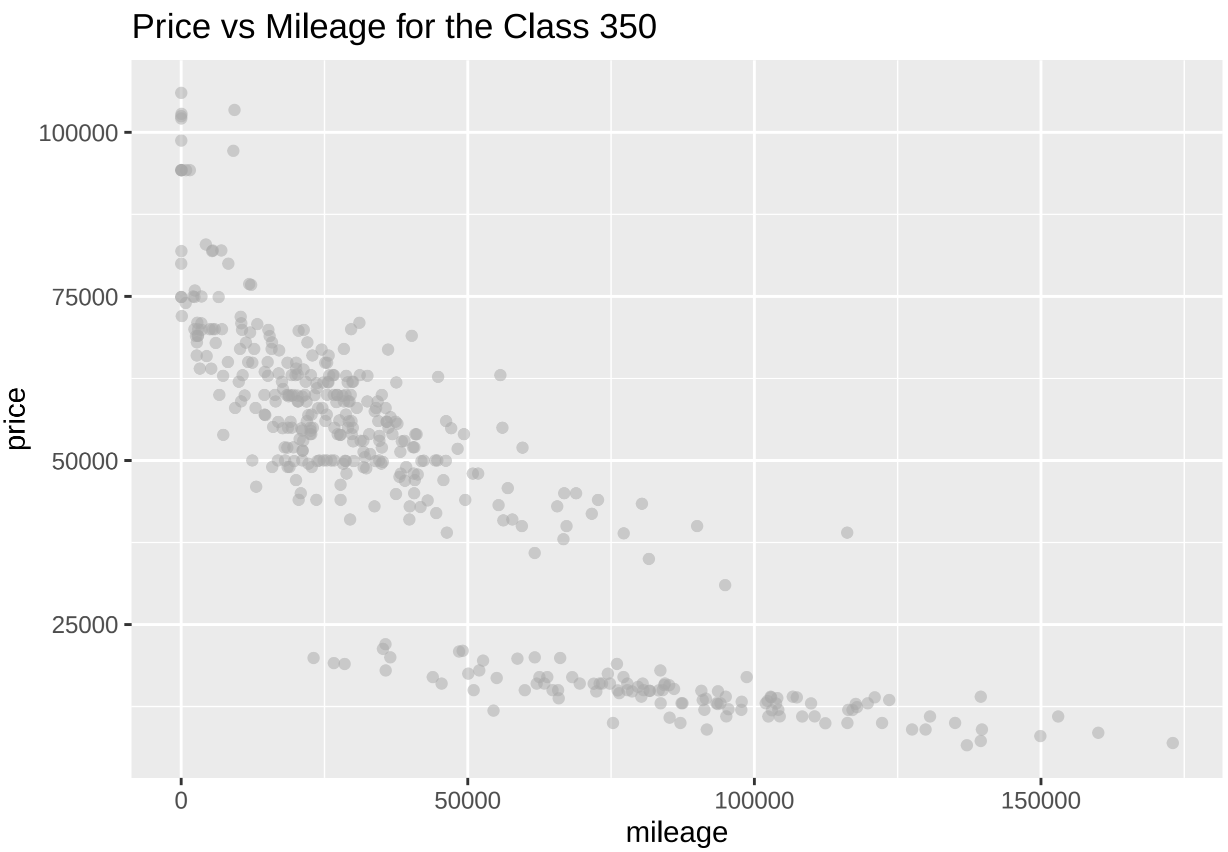


```
N = nrow(Class65AMG)
N_train = floor(0.8*N)
train_ind = sort(sample.int(N, N_train, replace=FALSE))

D_all = Class65AMG; D_all$set = ’test’; D_all$set[train_ind] = ’train’
D_train = Class65AMG[train_ind,]
D_test = Class65AMG[-train_ind,]

y_train = D_train$price
y_test = D_test$price
X_train = data.frame(mileage=jitter(D_train$mileage))
X_test = data.frame(mileage=jitter(D_test$mileage))


################RMSE Out-sample

k_grid = unique(round(exp(seq(log(450), log(2), length=100))))
rmse_grid_out = foreach(k = k_grid, .combine=’c’) %do% {
  knn_model = knnreg(price ~ mileage, data=D_train, k = k)
  modelr::rmse(knn_model, D_test)
}

rmse_grid_out = data.frame(K = k_grid, RMSE = rmse_grid_out)

p_out = ggplot(data=rmse_grid_out) +
  theme_bw(base_size = 10) +
  geom_path(aes(x=K, y=RMSE, color=’testset’), size=0.5) +
  scale_x_continuous(trans=revlog_trans(base = 10))


###########RMSE In-sample
k_grid = unique(round(exp(seq(log(450), log(2), length=100))))

rmse_grid_in = foreach(k = k_grid, .combine=’c’) %do% {
  knn_model = knnreg(price ~ mileage, data=D_train, k = k)
  modelr::rmse(knn_model, D_train)
}

revlog_trans <- function(base = exp(1)) {
  require(scales)
  ## Define the desired transformation.
  trans <- function(x){
    -log(x, base)
  }
  ## Define the reverse of the desired transformation
  inv <- function(x){
    base^(-x)
  }
  ## Creates the transformation
  scales::trans_new(paste("revlog-", base, sep = ""),
                    trans,
                    inv,  ## The reverse of the transformation
                    log_breaks(base = base), ## default way to define the scale breaks
                    domain = c(1e-100, Inf)
  )
}

rmse_grid_in = data.frame(K = k_grid, RMSE = rmse_grid_in)

######### Graph both

p_out = ggplot(data=rmse_grid_out) +
  theme_bw(base_size = 10) +
  geom_path(aes(x=K, y=RMSE, color=’testset’), size=0.5) +
  scale_x_continuous(trans=revlog_trans(base = 10))

ind_best = which.min(rmse_grid_out$RMSE)
k_best = k_grid[ind_best]

p_out + geom_path(data=rmse_grid_in, aes(x=K, y=RMSE, color=’trainset’),size=0.5) +
  scale_colour_manual(name="RMSE",
                      values=c(testset="black", trainset="grey")) +
  geom_vline(xintercept=k_best, color=’darkgreen’, size=1) +
  labs(title = ’RMSE versus K, optimal value of K’, subtitle = k_best)
```

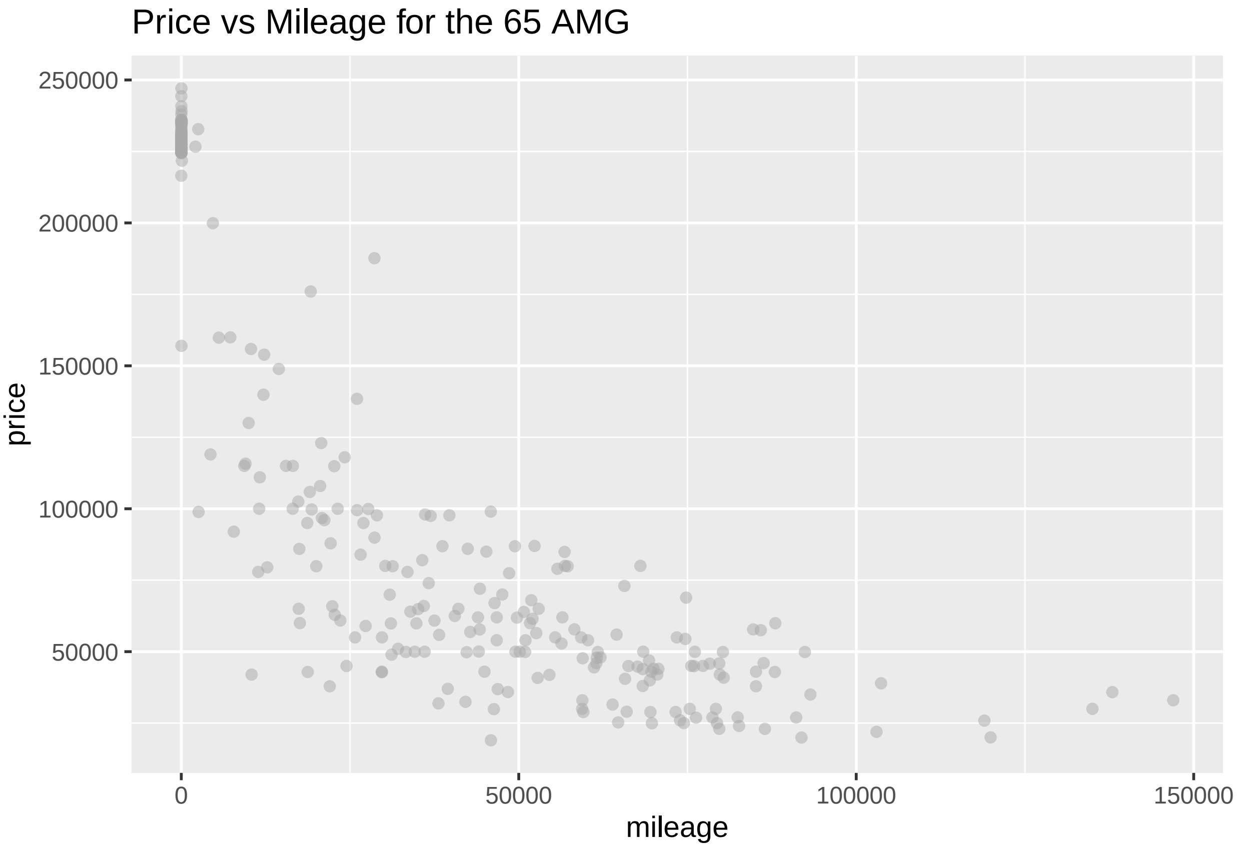


It might seem surprising that the optimal K value for the 65 AMG trim is 14, significantly different than what we have found for the Class 350. The reason for this, in my opinion, is that it has to do with the distribution of the data points. For example, when we look at the scatter plot for the Class 350, we can see that there is this sort of gap in the price range, seeming to create two different groups of cars. Whereas on the other hand, when we look at the 65 AMG data point distribution, there is a more “continuous” flow of the data points. Consequently, this means that in order to bridge the gap in prices for the Class 350, it has to take the average of more data points, as it has to make up for the lack of information.

```
Class350 <- sclass %>%
  filter(trim==350)
g0 = ggplot(data = Class350) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 +
  labs(title = 'Price vs Mileage for the Class 350')
```



```
Class65AMG <- sclass %>%
  filter(trim=='65 AMG')
g0 = ggplot(data = Class65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey', alpha=0.5)
g0 +
  labs(title = 'Price vs Mileage for the 65 AMG')
```


Price vs Mileage for the 65 AMG
price
250000
200000
150000
100000
50000
mileage
0
50000
100000
150000