HW08

Doug Nychka

2024-03-29

Setup for Audi A4 data.

Adjust the directory path to your laptop.

```
setwd("~/Dropbox/Home/Teaching/MATH531/MATH-531/MATH531S2024/Assignments")
```

Reformat/Wrangle Audi data

```
load("AudiA4.rda" )
head( AudiA4)
```

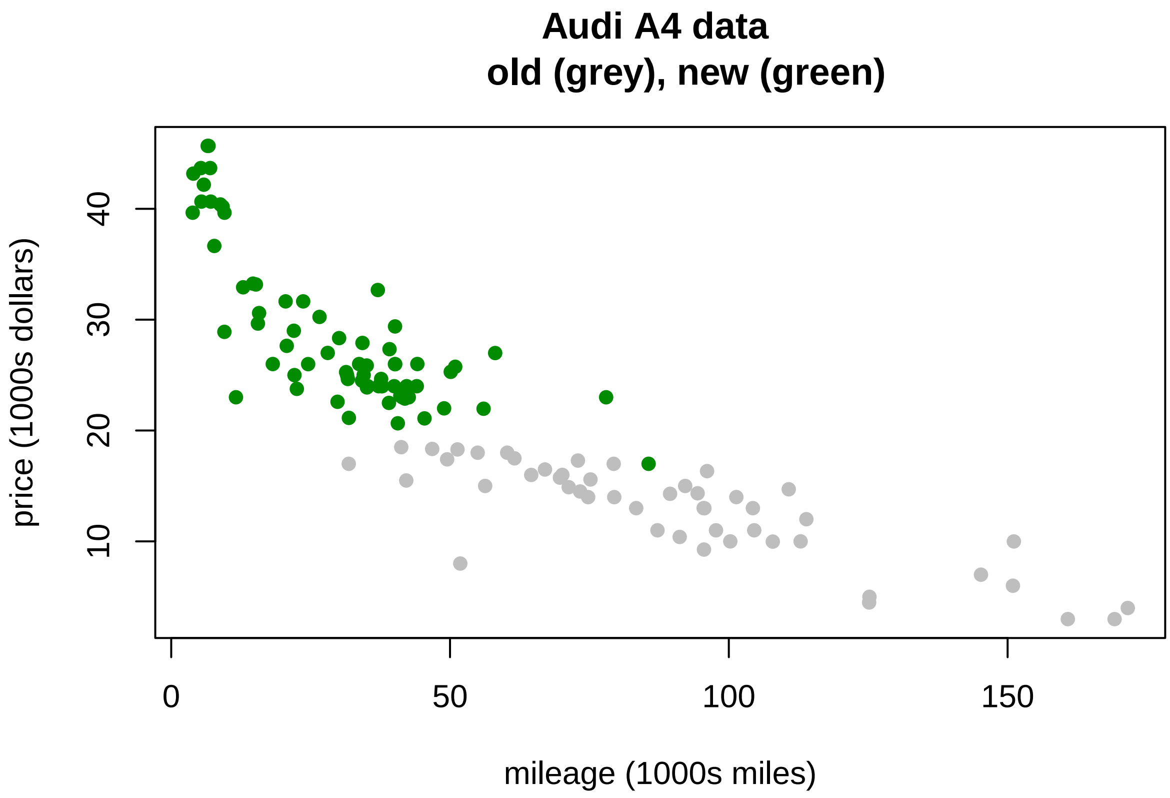
```
##     year price mileage distance
## 58  2020 39649    3848       29
## 145 2020 43175    3962        7
## 10  2020 43675    5316        7
## 52  2020 40649    5417       29
## 143 2020 42175    5846        7
## 9   2020 45675    6539        7
```

```
#convenient data frame for this HW
mileage<- AudiA4$mileage/1000
price<- AudiA4$price/1000
old<- ifelse( AudiA4$year<= 2016,1,0)
new<- ifelse( AudiA4$year > 2016,1,0)
age<- as.factor( ifelse( AudiA4$year<= 2016,"old","new"))
HW08AudiA4<- data.frame( price= price, mileage=mileage, old=old,
                         new=new,
                         age=age)
head( HW08AudiA4)
```

```
##    price mileage old new age
## 1 39.649   3.848   0   1 new
## 2 43.175   3.962   0   1 new
## 3 43.675   5.316   0   1 new
## 4 40.649   5.417   0   1 new
## 5 42.175   5.846   0   1 new
## 6 45.675   6.539   0   1 new
```

A plot of the data – always a good idea even for a "theory" class.

```
plot( HW08AudiA4$mileage, HW08AudiA4$price,
      col= ifelse(HW08AudiA4$old ==1, "grey", "green4"),
      pch=16,
      xlab="mileage (1000s miles)", ylab="price (1000s dollars)")
title("Audi A4 data
      old (grey), new (green) ")
```



Getting started on Problem 4c)

Example code for 90/10 cross-validation based on random partitions of the data.

```
n<- length( price)
M<- 1000 # number of random partitions.
CVA<- rep(NA, M)
CVB<- rep(NA, M)
set.seed( 531)
# find RMSE for out of sample prediction.
# fit on 90% and predit on 10% omitted
for( k in 1:M){
  ind90<- sample( 1:n, size= round(n*.9), replace=FALSE)

  data90<-HW08AudiA4[ind90,]
  data10<- HW08AudiA4[-ind90, ] # this is a handy way to _exclude_ indices but very idomatic!
    fitA<- lm( price ~  mileage + I(mileage^2) + I(mileage^3),
                 data=data90)
  predictA<- predict( fitA, newdata= data10)
  CVA[k] <-  sqrt(mean( (data10$price - predictA)^2))
  fitB<- lm( price ~  mileage + old,
             data=data90)
  predictB<- predict( fitB, newdata= data10)
  CVB[k] <-  sqrt(mean( (data10$price - predictB)^2))
}
```

Part of the problem is for you to interpret CVA and CVB and figure what to do with them!