02-exercises

Christopher Brown

April 13, 2016

This exercise uses the **Fuel Economy** data set from the **AppliedPredicitiveModeling** package.

Note: The following will set-up your environment for this exercise. If you get an error stating that the packages have not been found, you need to install those packages.

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

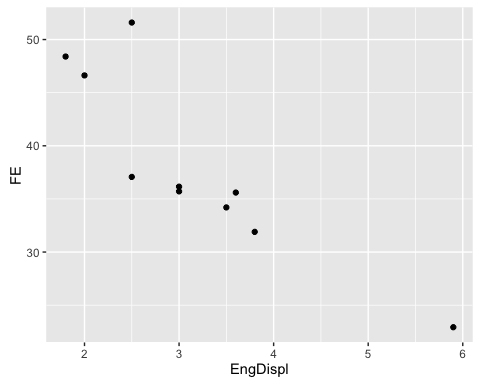
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Exercise 1

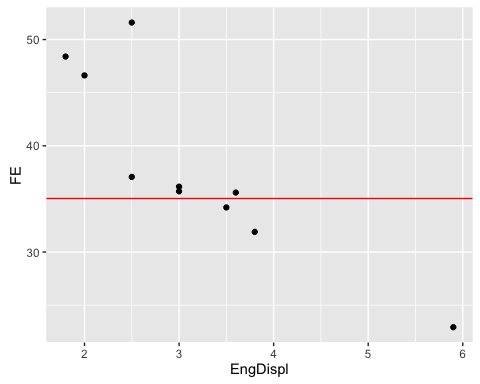
Hint: See ?cars2010

* After the **Fuel Economy** data is loaded, combine three data sets into one data set. (Note: The name dat is very often used in these situations, data is a reserved R word.)

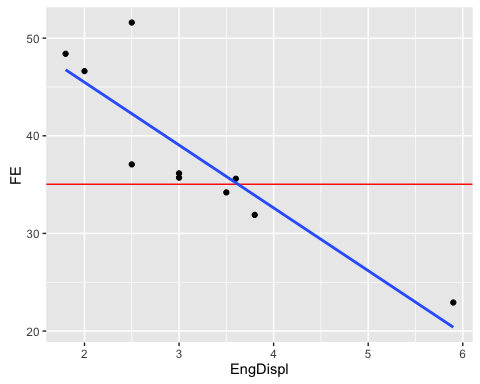
dat3years <- rbind(cars2010,cars2011,cars2012)  
  
naive\_guess = mean(dat3years$FE)  
  
set.seed(314)  
  
samp <- dat3years %>% dplyr::sample\_n(10) # you can also do just sample\_n(10) since no conflicts with deep lyer package  
  
samp %>% ggplot(aes(x=EngDispl, y=FE) ) + geom\_point()



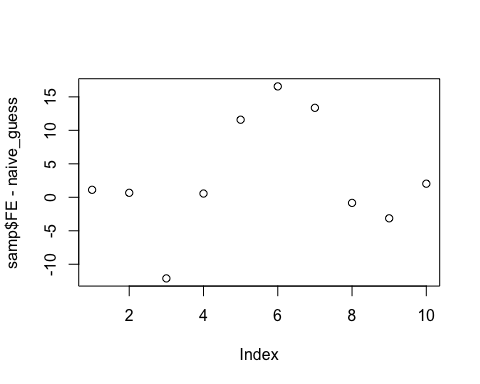
#Naive guess (red line) is just a good place to start, always need a good place to start  
samp %>% ggplot(aes(x=EngDispl, y=FE) ) + geom\_point() + geom\_hline(yintercept=naive\_guess, color="red")



#graph showing naive guess and linear model  
samp %>% ggplot(aes(x=EngDispl, y=FE) ) + geom\_point() + geom\_hline(yintercept=naive\_guess, color="red") + stat\_smooth(method = "lm", se=FALSE)

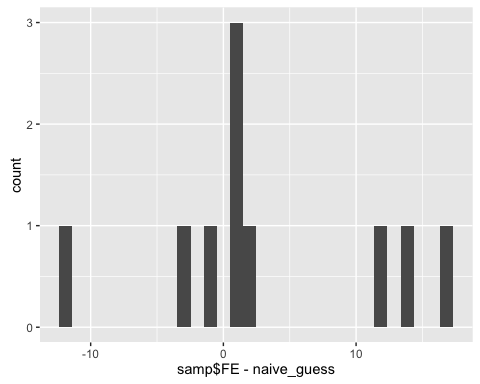


plot(samp$FE - naive\_guess)



qplot(samp$FE - naive\_guess)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#dat3years$FE  
  
(dat3years$FE - naive\_guess)^2 %>% mean %>% sqrt

## [1] 8.096176

fit.dat3years <- lm(FE ~ EngDispl, data = dat3years)  
  
fit.dat3years

##   
## Call:  
## lm(formula = FE ~ EngDispl, data = dat3years)  
##   
## Coefficients:  
## (Intercept) EngDispl   
## 51.840 -4.792

fit.dat3years <- lm(FE ~ EngDispl + NumCyl, data = dat3years)  
  
fit.dat3years

##   
## Call:  
## lm(formula = FE ~ EngDispl + NumCyl, data = dat3years)  
##   
## Coefficients:  
## (Intercept) EngDispl NumCyl   
## 52.6096 -4.1561 -0.5015

fit.dat3years <- lm(FE ~ EngDispl + NumCyl + NumGears, data = dat3years)  
  
fit.dat3years

##   
## Call:  
## lm(formula = FE ~ EngDispl + NumCyl + NumGears, data = dat3years)  
##   
## Coefficients:  
## (Intercept) EngDispl NumCyl NumGears   
## 52.75736 -4.16766 -0.48663 -0.03659

fit.dat3years <- lm(FE ~ EngDispl + CarlineClassDesc + EngDispl, data = dat3years)  
  
fit.dat3years

##   
## Call:  
## lm(formula = FE ~ EngDispl + CarlineClassDesc + EngDispl, data = dat3years)  
##   
## Coefficients:  
## (Intercept)   
## 47.3513   
## EngDispl   
## -4.2855   
## CarlineClassDesc2Seaters   
## 3.3389   
## CarlineClassDescCompactCars   
## 5.7699   
## CarlineClassDescLargeCars   
## 3.9337   
## CarlineClassDescMidsizeCars   
## 5.4664   
## CarlineClassDescMinicompactCars   
## 4.8408   
## CarlineClassDescSmallPickupTrucks2WD   
## -1.4834   
## CarlineClassDescSmallPickupTrucks4WD   
## -1.8652   
## CarlineClassDescSmallStationWagons   
## 2.7078   
## CarlineClassDescSpecialPurposeVehicleminivan2WD   
## 1.8885   
## CarlineClassDescSpecialPurposeVehicleSUV2WD   
## 1.7443   
## CarlineClassDescSpecialPurposeVehicleSUV4WD   
## -0.7989   
## CarlineClassDescStandardPickupTrucks2WD   
## 1.1908   
## CarlineClassDescStandardPickupTrucks4WD   
## -0.8495   
## CarlineClassDescSubcompactCars   
## 4.0061   
## CarlineClassDescVansCargoTypes   
## -1.4134   
## CarlineClassDescVansPassengerType   
## -1.9240

fit.dat3years %>% summary()

##   
## Call:  
## lm(formula = FE ~ EngDispl + CarlineClassDesc + EngDispl, data = dat3years)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.5863 -2.5786 -0.3323 2.0675 24.5366   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 47.35133 1.08855  
## EngDispl -4.28549 0.09444  
## CarlineClassDesc2Seaters 3.33895 1.13930  
## CarlineClassDescCompactCars 5.76985 1.09449  
## CarlineClassDescLargeCars 3.93368 1.14347  
## CarlineClassDescMidsizeCars 5.46636 1.10046  
## CarlineClassDescMinicompactCars 4.84079 1.17630  
## CarlineClassDescSmallPickupTrucks2WD -1.48338 1.26635  
## CarlineClassDescSmallPickupTrucks4WD -1.86522 1.37416  
## CarlineClassDescSmallStationWagons 2.70785 1.15035  
## CarlineClassDescSpecialPurposeVehicleminivan2WD 1.88848 1.41360  
## CarlineClassDescSpecialPurposeVehicleSUV2WD 1.74434 1.10760  
## CarlineClassDescSpecialPurposeVehicleSUV4WD -0.79885 1.09391  
## CarlineClassDescStandardPickupTrucks2WD 1.19076 1.27855  
## CarlineClassDescStandardPickupTrucks4WD -0.84945 1.26361  
## CarlineClassDescSubcompactCars 4.00608 1.10918  
## CarlineClassDescVansCargoTypes -1.41338 1.40369  
## CarlineClassDescVansPassengerType -1.92396 1.46703  
## t value Pr(>|t|)   
## (Intercept) 43.500 < 2e-16 \*\*\*  
## EngDispl -45.377 < 2e-16 \*\*\*  
## CarlineClassDesc2Seaters 2.931 0.003436 \*\*   
## CarlineClassDescCompactCars 5.272 1.56e-07 \*\*\*  
## CarlineClassDescLargeCars 3.440 0.000598 \*\*\*  
## CarlineClassDescMidsizeCars 4.967 7.61e-07 \*\*\*  
## CarlineClassDescMinicompactCars 4.115 4.09e-05 \*\*\*  
## CarlineClassDescSmallPickupTrucks2WD -1.171 0.241643   
## CarlineClassDescSmallPickupTrucks4WD -1.357 0.174883   
## CarlineClassDescSmallStationWagons 2.354 0.018711 \*   
## CarlineClassDescSpecialPurposeVehicleminivan2WD 1.336 0.181784   
## CarlineClassDescSpecialPurposeVehicleSUV2WD 1.575 0.115505   
## CarlineClassDescSpecialPurposeVehicleSUV4WD -0.730 0.465343   
## CarlineClassDescStandardPickupTrucks2WD 0.931 0.351834   
## CarlineClassDescStandardPickupTrucks4WD -0.672 0.501537   
## CarlineClassDescSubcompactCars 3.612 0.000315 \*\*\*  
## CarlineClassDescVansCargoTypes -1.007 0.314149   
## CarlineClassDescVansPassengerType -1.311 0.189913   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.212 on 1429 degrees of freedom  
## Multiple R-squared: 0.7328, Adjusted R-squared: 0.7296   
## F-statistic: 230.5 on 17 and 1429 DF, p-value: < 2.2e-16

#dot means use all variables   
fit.dat3years <- lm(FE ~ . + EngDispl, data = dat3years)  
  
fit.dat3years

##   
## Call:  
## lm(formula = FE ~ . + EngDispl, data = dat3years)  
##   
## Coefficients:  
## (Intercept)   
## 57.23884   
## EngDispl   
## -2.34608   
## NumCyl   
## -1.08024   
## TransmissionA4   
## -6.85302   
## TransmissionA5   
## -5.44388   
## TransmissionA6   
## -3.56074   
## TransmissionA7   
## -2.96101   
## TransmissionAM6   
## -5.54718   
## TransmissionAM7   
## -6.46621   
## TransmissionAV   
## -4.87101   
## TransmissionAVS6   
## -7.71471   
## TransmissionM5   
## -6.08294   
## TransmissionM6   
## -5.82397   
## TransmissionS4   
## -9.49087   
## TransmissionS5   
## -7.02958   
## TransmissionS6   
## -4.41966   
## TransmissionS7   
## -4.30174   
## TransmissionS8   
## -1.09346   
## AirAspirationMethodSupercharged   
## -1.07233   
## AirAspirationMethodTurbocharged   
## -0.49004   
## NumGears   
## -0.54137   
## TransLockup   
## -0.86099   
## TransCreeperGear   
## -0.53964   
## DriveDescFourWheelDrive   
## -0.17575   
## DriveDescParttimeFourWheelDrive   
## 0.08028   
## DriveDescTwoWheelDriveFront   
## 5.48985   
## DriveDescTwoWheelDriveRear   
## 1.52322   
## IntakeValvePerCyl   
## -0.88678   
## ExhaustValvesPerCyl   
## -1.07324   
## CarlineClassDesc2Seaters   
## 3.69574   
## CarlineClassDescCompactCars   
## 4.50549   
## CarlineClassDescLargeCars   
## 3.46899   
## CarlineClassDescMidsizeCars   
## 4.22037   
## CarlineClassDescMinicompactCars   
## 4.09843   
## CarlineClassDescSmallPickupTrucks2WD   
## -1.23092   
## CarlineClassDescSmallPickupTrucks4WD   
## -0.28463   
## CarlineClassDescSmallStationWagons   
## 2.73308   
## CarlineClassDescSpecialPurposeVehicleminivan2WD   
## -2.37816   
## CarlineClassDescSpecialPurposeVehicleSUV2WD   
## -1.14331   
## CarlineClassDescSpecialPurposeVehicleSUV4WD   
## 0.39161   
## CarlineClassDescStandardPickupTrucks2WD   
## -0.53602   
## CarlineClassDescStandardPickupTrucks4WD   
## -0.99735   
## CarlineClassDescSubcompactCars   
## 3.85189   
## CarlineClassDescVansCargoTypes   
## -3.30398   
## CarlineClassDescVansPassengerType   
## -4.51100   
## VarValveTiming   
## 0.21536   
## VarValveLift   
## 1.07734

#sample 10 CarlineClassDesc  
dat3years %>% select(CarlineClassDesc) %>% sample\_n(10)

## CarlineClassDesc  
## 1622 LargeCars  
## 1255 SubcompactCars  
## 1438 CompactCars  
## 1396 CompactCars  
## 1817 StandardPickupTrucks2WD  
## 1653 LargeCars  
## 2129 SpecialPurposeVehicleSUV4WD  
## 1837 StandardPickupTrucks4WD  
## 1143 2Seaters  
## 1573 MidsizeCars

#print table count for each CarlineClassDesc type  
dat3years %>% select(CarlineClassDesc) %>% table

## .  
## Other 2Seaters   
## 16 98   
## CompactCars LargeCars   
## 199 98   
## MidsizeCars MinicompactCars   
## 175 65   
## SmallPickupTrucks2WD SmallPickupTrucks4WD   
## 36 23   
## SmallStationWagons SpecialPurposeVehicleminivan2WD   
## 84 20   
## SpecialPurposeVehicleSUV2WD SpecialPurposeVehicleSUV4WD   
## 154 218   
## StandardPickupTrucks2WD StandardPickupTrucks4WD   
## 36 39   
## SubcompactCars VansCargoTypes   
## 146 22   
## VansPassengerType   
## 18

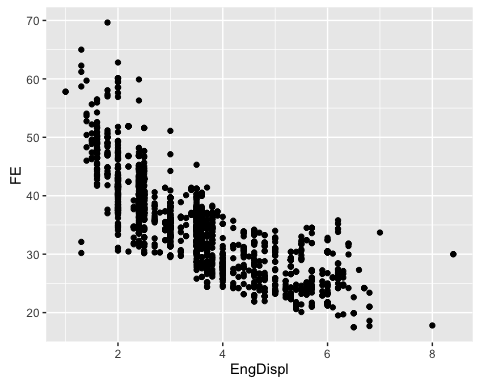
* What is a good "naive guess" of FE? Show your work

dat3years <- rbind(cars2010,cars2011,cars2012)  
  
naive\_guess = mean(dat3years$FE)  
  
naive\_guess

## [1] 35.03823

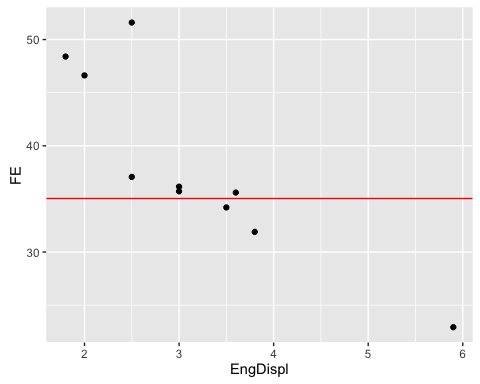
* plot FE (Fuel Econonomy) vs. EngDisp. Plot the naive guess.

# ... ggplot2   
  
dat3years %>% ggplot(aes(x=EngDispl, y=FE) ) + geom\_point()



* Sample 10 observations from dat
* Plot this data. Add a line for the naive\_guess.

set.seed(314)   
  
# Sample   
  
samp <- dat3years %>% dplyr::sample\_n(10)   
  
#Naive guess (red line) is just a good place to start, always need a good place to start  
samp %>% ggplot(aes(x=EngDispl, y=FE) ) + geom\_point() + geom\_hline(yintercept=naive\_guess, color="red")



## Exercise 2:

Write a loss functions for calculating:

* Root Mean Square Error
* Mean Absolute Error
* Median Absolute Error

All functions should accept two arguments:

rmse <- function(y,yhat) {  
  
 ( y - yhat )^2 %>% mean %>% sqrt   
}  
  
mae <- function(y, yhat) {  
   
 abs( y - yhat ) %>% mean()  
}  
  
medae <- function(y, yhat) {   
   
 abs( y - yhat ) %>% median()  
}

Use these functions to evaluate the loss/performance of: - the naive guess

## Exercise 3: Linear Model and Model Performance

* Use lm to create a linear model fitting the relationship between FE and EngDispl for the cars2010 data set

fit.2010 <- lm( FE ~ EngDispl, data=cars2010 )

* Use your functions to evaluate the training error
* Use your model to: -- predict the FE for 2011. What is the RMSE errors associated with the predictions. -- predict the FE for 2012. What is the RMSE errors associated with the predictions.

#Predict FE for 2010, 2011, 2012 using lm from 2010  
y.2010 <- predict( fit.2010, data=cars2010 )  
y.2011 <- predict( fit.2010, data=cars2011 )  
y.2012 <- predict( fit.2010, data=cars2012 )  
  
#Calculate RMSE error  
rmse.2010 <- rmse( cars2010$FE,y.2010)  
rmse.2011 <- rmse( cars2011$FE,y.2011)

## Warning in y - yhat: longer object length is not a multiple of shorter  
## object length

rmse.2012 <- rmse( cars2012$FE,y.2012)

## Warning in y - yhat: longer object length is not a multiple of shorter  
## object length

# DO NOT EDIT   
rmse.2010

## [1] 4.620076

rmse.2011

## [1] 11.33028

rmse.2012

## [1] 12.94582

## Exercise 4:

* Model the fuel economy (FE) as a function of EngDispl, NumCyl and VarValve using the cars2011 data set.
* Provide betas

fit.2011 <- lm( FE ~ EngDispl + NumCyl + VarValveTiming, data=cars2011 )   
  
summary(fit.2011)

##   
## Call:  
## lm(formula = FE ~ EngDispl + NumCyl + VarValveTiming, data = cars2011)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.687 -2.768 -0.960 2.279 19.124   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 52.6644 1.4701 35.823 < 2e-16 \*\*\*  
## EngDispl -3.9056 0.5246 -7.445 1.71e-12 \*\*\*  
## NumCyl -1.1102 0.4268 -2.601 0.00987 \*\*   
## VarValveTiming 3.5937 0.8862 4.055 6.76e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.828 on 241 degrees of freedom  
## Multiple R-squared: 0.7283, Adjusted R-squared: 0.725   
## F-statistic: 215.4 on 3 and 241 DF, p-value: < 2.2e-16

#install.packages("lm.beta")  
#library(lm.beta)  
  
#fit.2011.beta <- lm.beta(fit.2011)  
  
#summary(fit.2011.beta)  
#coef(fit.2011.beta)