TFM\_Modelado

2024-02-27

# Carga de datos.  
getwd()

## [1] "C:/Users/34682/Desktop/TFM/Posibles\_datasets/vehicles"

setwd("C:/Users/34682/Desktop/TFM/Posibles\_datasets/vehicles/")  
v<-read.csv('Australian vehicle Prices Clean.csv')  
  
  
# Validación cruzada 4-fold.  
# Separación conjuntos de entrenamiento, validación y test.  
set.seed(9)  
train\_ind <- sample(seq\_len(nrow(v)), size =floor(0.9\*nrow(v)))  
v\_train <- v[train\_ind, ]  
v\_test <- v[-train\_ind, ]  
valid\_cros\_ind\_1 <- sample(seq\_len(nrow(v\_train)),   
 size =floor(0.5\*nrow(v\_train)))  
v\_train\_aux1 <- v\_train[valid\_cros\_ind\_1, ]  
v\_train\_aux2 <- v\_train[-valid\_cros\_ind\_1, ]  
valid\_cros\_ind\_2 <- sample(seq\_len(nrow(v\_train\_aux1)),   
 size =floor(0.5\*nrow(v\_train\_aux1)))  
valid\_cros\_ind\_3 <- sample(seq\_len(nrow(v\_train\_aux2)),   
 size =floor(0.5\*nrow(v\_train\_aux2)))  
v\_train\_1 <- v\_train\_aux1[valid\_cros\_ind\_2, ]  
v\_train\_2 <- v\_train\_aux1[-valid\_cros\_ind\_2, ]  
v\_train\_3 <- v\_train\_aux2[valid\_cros\_ind\_3, ]  
v\_train\_4 <- v\_train\_aux2[-valid\_cros\_ind\_3, ]  
v\_train\_i1 <- rbind(v\_train\_2, v\_train\_3, v\_train\_4)  
v\_train\_i2 <- rbind(v\_train\_1, v\_train\_3, v\_train\_4)  
v\_train\_i3 <- rbind(v\_train\_1, v\_train\_2, v\_train\_4)  
v\_train\_i4 <- rbind(v\_train\_1, v\_train\_2, v\_train\_3)  
  
  
# Normal con función de enlace la identidad.  
mg1\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "identity"),v\_train\_i1)  
pred\_val <-predict(mg1\_1, newdata = v\_train\_1, ty="response")  
mg1\_pr1\_val <- mean(abs(v\_train\_1$Price - pred\_val))  
mg1\_pr1\_tr <- mean(abs(v\_train\_i1$Price - mg1\_1$fitted.values))  
rC1\_1 <- 1- (mg1\_1$deviance/mg1\_1$null.deviance)   
  
mg1\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "identity"),v\_train\_i2)  
pred\_val <-predict(mg1\_2, newdata = v\_train\_2, ty="response")  
mg1\_pr2\_val <- mean(abs(v\_train\_2$Price - pred\_val))  
mg1\_pr2\_tr <- mean(abs(v\_train\_i2$Price - mg1\_2$fitted.values))  
rC1\_2 <- 1- (mg1\_2$deviance/mg1\_2$null.deviance)   
  
mg1\_3 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "identity"),v\_train\_i3)  
pred\_val <-predict(mg1\_3, newdata = v\_train\_3, ty="response")  
mg1\_pr3\_val <- mean(abs(v\_train\_3$Price - pred\_val))  
mg1\_pr3\_tr <- mean(abs(v\_train\_i3$Price - mg1\_3$fitted.values))  
rC1\_3 <- 1- (mg1\_3$deviance/mg1\_3$null.deviance)   
  
mg1\_4 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "identity"),v\_train\_i4)  
pred\_val <-predict(mg1\_4, newdata = v\_train\_4, ty="response")  
mg1\_pr4\_val <- mean(abs(v\_train\_4$Price - pred\_val))  
mg1\_pr4\_tr <- mean(abs(v\_train\_i4$Price - mg1\_4$fitted.values))  
rC1\_4 <- 1- (mg1\_4$deviance/mg1\_4$null.deviance)   
  
#MAE\_v  
mg1\_pr\_val <- mean(c(mg1\_pr1\_val, mg1\_pr2\_val, mg1\_pr3\_val, mg1\_pr4\_val))  
#MAE\_train  
mg1\_pr\_tr <- mean(c(mg1\_pr1\_tr, mg1\_pr2\_tr, mg1\_pr3\_tr, mg1\_pr4\_tr))  
#mean(pseudo R^2)  
rC1 <- mean(c(rC1\_1, rC1\_2, rC1\_3, rC1\_4))  
c(mg1\_pr\_val, mg1\_pr\_tr, rC1)

## [1] 8598.3121663 8586.7025402 0.6511772

#Normal con función de enlace log  
mg2\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "log"),v\_train\_i1)  
pred\_val <-predict(mg2\_1, newdata = v\_train\_1, ty="response")  
mg2\_pr1\_val <- mean(abs(v\_train\_1$Price - pred\_val))  
mg2\_pr1\_tr <- mean(abs(v\_train\_i1$Price - mg2\_1$fitted.values))  
rC2\_1 <- 1- (mg2\_1$deviance/mg2\_1$null.deviance)   
  
mg2\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders   
 + BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "log"),v\_train\_i2)  
pred\_val <-predict(mg2\_2, newdata = v\_train\_2, ty="response")  
mg2\_pr2\_val <- mean(abs(v\_train\_2$Price - pred\_val))  
mg2\_pr2\_tr <- mean(abs(v\_train\_i2$Price - mg2\_2$fitted.values))  
rC2\_2 <- 1- (mg2\_2$deviance/mg2\_2$null.deviance)   
  
mg2\_3 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "log"),v\_train\_i3)  
pred\_val <-predict(mg2\_3, newdata = v\_train\_3, ty="response")  
mg2\_pr3\_val <- mean(abs(v\_train\_3$Price - pred\_val))  
mg2\_pr3\_tr <- mean(abs(v\_train\_i3$Price - mg2\_3$fitted.values))  
rC2\_3 <- 1- (mg2\_3$deviance/mg2\_3$null.deviance)   
  
mg2\_4 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "log"),v\_train\_i4)  
pred\_val <-predict(mg2\_4, newdata = v\_train\_4, ty="response")  
mg2\_pr4\_val <- mean(abs(v\_train\_4$Price - pred\_val))  
mg2\_pr4\_tr <- mean(abs(v\_train\_i4$Price - mg2\_4$fitted.values))  
rC2\_4 <- 1- (mg2\_4$deviance/mg2\_4$null.deviance)   
  
#MAE\_v  
mg2\_pr\_val <- mean(c(mg2\_pr1\_val, mg2\_pr2\_val, mg2\_pr3\_val, mg2\_pr4\_val))  
#MAE\_train  
mg2\_pr\_tr <- mean(c(mg2\_pr1\_tr, mg2\_pr2\_tr, mg2\_pr3\_tr, mg2\_pr4\_tr))  
#mean(pseudo R^2)  
rC2 <- mean(c(rC2\_1, rC2\_2, rC2\_3, rC2\_4))  
c(mg2\_pr\_val, mg2\_pr\_tr, rC2)

## [1] 7307.4239934 7287.8358525 0.7209465

#Normal con función de enlace inversa  
mg3\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "inverse"),v\_train\_i1)  
pred\_val <-predict(mg3\_1, newdata = v\_train\_1, ty="response")  
mg3\_pr1\_val <- mean(abs(v\_train\_1$Price - pred\_val))  
mg3\_pr1\_tr <- mean(abs(v\_train\_i1$Price - mg3\_1$fitted.values))  
rC3\_1 <- 1- (mg3\_1$deviance/mg3\_1$null.deviance)   
  
mg3\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType  
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "inverse"),v\_train\_i2)  
pred\_val <-predict(mg3\_2, newdata = v\_train\_2, ty="response")  
mg3\_pr2\_val <- mean(abs(v\_train\_2$Price - pred\_val))  
mg3\_pr2\_tr <- mean(abs(v\_train\_i2$Price - mg3\_2$fitted.values))  
rC3\_2 <- 1- (mg3\_2$deviance/mg3\_2$null.deviance)   
  
mg3\_3 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "inverse"),v\_train\_i3)  
pred\_val <-predict(mg3\_3, newdata = v\_train\_3, ty="response")  
mg3\_pr3\_val <- mean(abs(v\_train\_3$Price - pred\_val))  
mg3\_pr3\_tr <- mean(abs(v\_train\_i3$Price - mg3\_3$fitted.values))  
rC3\_3 <- 1- (mg3\_3$deviance/mg3\_3$null.deviance)   
  
mg3\_4 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType   
 + FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= gaussian(link= "inverse"),v\_train\_i4)  
pred\_val <-predict(mg3\_4, newdata = v\_train\_4, ty="response")  
mg3\_pr4\_val <- mean(abs(v\_train\_4$Price - pred\_val))  
mg3\_pr4\_tr <- mean(abs(v\_train\_i4$Price - mg3\_4$fitted.values))  
rC3\_4 <- 1- (mg3\_4$deviance/mg3\_4$null.deviance)   
#MAE\_v  
mg3\_pr\_val <- mean(c(mg3\_pr1\_val, mg3\_pr2\_val, mg3\_pr3\_val, mg3\_pr4\_val))  
#MAE\_train  
mg3\_pr\_tr <- mean(c(mg3\_pr1\_tr, mg3\_pr2\_tr, mg3\_pr3\_tr, mg3\_pr4\_tr))  
#mean(pseudo R^2)  
rC3 <- mean(c(rC3\_1, rC3\_2, rC3\_3, rC3\_4))  
c(mg3\_pr\_val, mg3\_pr\_tr, rC3)

## [1] 8161.7934057 8125.1032348 0.6825537

#Gamma con función de enlace la identidad  
# mg4\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
#DriveType  
# + FuelType + FuelConsumption + Kilometres + Cylinders + BodyType   
# + Doors + Seats + DollarAustralian + PriceIndex ,  
# family= Gamma(link= "identity"),v\_train\_i1)  
# El algoritmo de estimación de parámetros no converge y   
# no se puede considerar dicho modelo.  
  
#Gamma con función de enlace log  
mg5\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType +   
 FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex ,  
 family= Gamma(link= "log"),v\_train\_i1)  
pred\_val <-predict(mg5\_1, newdata = v\_train\_1, ty="response")  
mg5\_pr1\_val <- mean(abs(v\_train\_1$Price - pred\_val))  
mg5\_pr1\_tr <- mean(abs(v\_train\_i1$Price - mg5\_1$fitted.values))  
rC5\_1 <- 1- (mg5\_1$deviance/mg5\_1$null.deviance)   
  
mg5\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType +   
 FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex ,  
 family= Gamma(link= "log"),v\_train\_i2)  
pred\_val <-predict(mg5\_2, newdata = v\_train\_2, ty="response")  
mg5\_pr2\_val <- mean(abs(v\_train\_2$Price - pred\_val))  
mg5\_pr2\_tr <- mean(abs(v\_train\_i2$Price - mg5\_2$fitted.values))  
rC5\_2 <- 1- (mg5\_2$deviance/mg5\_2$null.deviance)   
  
mg5\_3 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType +   
 FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex ,  
 family= Gamma(link= "log"),v\_train\_i3)  
pred\_val <-predict(mg5\_3, newdata = v\_train\_3, ty="response")  
mg5\_pr3\_val <- mean(abs(v\_train\_3$Price - pred\_val))  
mg5\_pr3\_tr <- mean(abs(v\_train\_i3$Price - mg5\_3$fitted.values))  
rC5\_3 <- 1- (mg5\_3$deviance/mg5\_3$null.deviance)   
  
mg5\_4 <- glm(Price~ Year + UsedOrNew + Transmission + Engine + DriveType  
 + FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex ,  
 family= Gamma(link= "log"),v\_train\_i4)  
pred\_val <-predict(mg5\_4, newdata = v\_train\_4, ty="response")  
mg5\_pr4\_val <- mean(abs(v\_train\_4$Price - pred\_val))  
mg5\_pr4\_tr <- mean(abs(v\_train\_i4$Price - mg5\_4$fitted.values))  
rC5\_4 <- 1- (mg5\_4$deviance/mg5\_4$null.deviance)   
  
#MAE\_v  
mg5\_pr\_val <- mean(c(mg5\_pr1\_val, mg5\_pr2\_val, mg5\_pr3\_val, mg5\_pr4\_val))  
#MAE\_train  
mg5\_pr\_tr <- mean(c(mg5\_pr1\_tr, mg5\_pr2\_tr, mg5\_pr3\_tr, mg5\_pr4\_tr))  
#mean(pseudo R^2)  
rC5 <- mean(c(rC5\_1, rC5\_2, rC5\_3, rC5\_4))  
c(mg5\_pr\_val, mg5\_pr\_tr, rC5)

## [1] 7306.0531022 7294.2724383 0.7671627

#Gamma con función de enlace inversa.  
# mg6\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
# DriveType + FuelType + FuelConsumption + Kilometres + Cylinders +  
# BodyType + Doors + Seats + DollarAustralian + PriceIndex ,  
# family= Gamma(link= "inverse"),v\_train\_i1)  
# El algoritmo de estimación de parámetros no converge y  
# no se puede considerar dicho modelo.  
  
#Inversa gaussiana con función de enlace identidad  
# mg7\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine   
# + DriveType   
# + FuelType + FuelConsumption + Kilometres + Cylinders + BodyType  
# + Doors +  
# Seats + DollarAustralian + PriceIndex ,  
# family= inverse.gaussian(link= "identity"),v\_train\_i1)  
# El algoritmo de estimación de parámetros no converge y   
# no se puede considerar dicho modelo.  
  
#Inversa gaussiana con función de enlace log  
# mg8\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +  
# DriveType +   
# FuelType + FuelConsumption + Kilometres + Cylinders + BodyType +   
# Doors +  
# Seats + DollarAustralian + PriceIndex ,  
# family= inverse.gaussian(link= "log"),v\_train\_i1)  
# pred\_val <-predict(mg8\_1, newdata = v\_train\_1, ty="response")  
# mg8\_pr1\_val <- mean(abs(v\_train\_1$Price - pred\_val))  
# mg8\_pr1\_tr <- mean(abs(v\_train\_i1$Price - mg8\_1$fitted.values))  
# rC8\_1 <- 1- (mg8\_1$deviance/mg8\_1$null.deviance)   
# mg8\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
# DriveType +   
# FuelType + FuelConsumption + Kilometres + Cylinders + BodyType   
# + Doors   
# + Seats + DollarAustralian + PriceIndex ,  
# family= inverse.gaussian(link= "log"),v\_train\_i2)  
# El algoritmo de estimación de parámetros no converge y   
# no se puede considerar dicho modelo.  
  
#Inversa gaussiana con función de enlace inversa  
# mg9\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
# DriveType +  
# FuelType + FuelConsumption + Kilometres + Cylinders + BodyType +  
# Doors +   
# Seats + DollarAustralian + PriceIndex ,  
# family= inverse.gaussian(link= "inverse"),v\_train\_i1)  
# El algoritmo de estimación de parámetros no converge y  
# no se puede considerar dicho modelo.  
  
#Inversa gaussiana con función de enlace inversa cuadrática  
# mg10\_1 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
# DriveType + FuelType + FuelConsumption + Kilometres + Cylinders   
# + BodyType  
# + Doors + Seats + DollarAustralian + PriceIndex ,  
# family= inverse.gaussian(link= "1/mu^2"),v\_train\_i1)  
# El algoritmo de estimación de parámetros no converge y   
# no se puede considerar dicho modelo.  
  
#Poisson con función de enlace identidad  
# mg11\_1 <- glm(as.integer(Price)~ Year + UsedOrNew + Transmission   
# + Engine   
# + DriveType + FuelType + FuelConsumption + Kilometres + Cylinders +   
# BodyType + Doors + Seats + DollarAustralian + PriceIndex ,  
# family= poisson(link= "identity"),v\_train\_i1)  
# El algoritmo de estimación de parámetros no converge y no   
# se puede considerar dicho modelo.  
  
#Poisson con función de enlace log  
mg12\_1 <- glm(as.integer(Price)~ Year + UsedOrNew + Transmission +   
 Engine +  
 DriveType + FuelType + FuelConsumption + Kilometres +   
 Cylinders + BodyType + Doors + Seats + DollarAustralian +  
 PriceIndex ,family= poisson(link= "log"),v\_train\_i1)  
pred\_val <-predict(mg12\_1, newdata = v\_train\_1, ty="response")  
mg12\_pr1\_val <- mean(abs(as.integer(v\_train\_1$Price) - round(pred\_val,0)))  
mg12\_pr1\_tr <- mean(abs(as.integer(v\_train\_i1$Price) -   
 round(mg12\_1$fitted.values,0)))  
rC12\_1 <- 1- (mg12\_1$deviance/mg12\_1$null.deviance)   
  
mg12\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
 DriveType +  
 FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= poisson(link= "log"),v\_train\_i2)  
pred\_val <-predict(mg12\_2, newdata = v\_train\_2, ty="response")  
mg12\_pr2\_val <- mean(abs(as.integer(v\_train\_2$Price) - round(pred\_val,0)))  
mg12\_pr2\_tr <- mean(abs(as.integer(v\_train\_i2$Price) -  
 round(mg12\_2$fitted.values,0)))  
rC12\_2 <- 1- (mg12\_2$deviance/mg12\_2$null.deviance)   
  
mg12\_3 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
 DriveType +   
 FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= poisson(link= "log"),v\_train\_i3)  
pred\_val <-predict(mg12\_3, newdata = v\_train\_3, ty="response")  
mg12\_pr3\_val <- mean(abs(as.integer(v\_train\_3$Price) - round(pred\_val,0)))  
mg12\_pr3\_tr <- mean(abs(as.integer(v\_train\_i3$Price) -   
 round(mg12\_3$fitted.values,0)))  
rC12\_3 <- 1- (mg12\_3$deviance/mg12\_3$null.deviance)   
  
mg12\_4 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
 DriveType +  
 FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= poisson(link= "log"),v\_train\_i4)  
pred\_val <-predict(mg12\_4, newdata = v\_train\_4, ty="response")  
mg12\_pr4\_val <- mean(abs(as.integer(v\_train\_4$Price) - round(pred\_val,0)))  
mg12\_pr4\_tr <- mean(abs(as.integer(v\_train\_i4$Price) -   
 round(mg12\_4$fitted.values,0)))  
rC12\_4 <- 1- (mg12\_4$deviance/mg12\_4$null.deviance)   
  
#MAE\_v  
mg12\_pr\_val <- mean(c(mg12\_pr1\_val, mg12\_pr2\_val, mg12\_pr3\_val,   
 mg12\_pr4\_val))  
#MAE\_train  
mg12\_pr\_tr <- mean(c(mg12\_pr1\_tr, mg12\_pr2\_tr, mg12\_pr3\_tr, mg12\_pr4\_tr))  
#mean(pseudo R^2)  
rC12 <- mean(c(rC12\_1, rC12\_2, rC12\_3, rC12\_4))  
c(mg12\_pr\_val, mg12\_pr\_tr, rC12)

## [1] 7245.3824468 7230.8703060 0.7595578

#Poisson con función de enlace sqrt  
mg13\_1 <- glm(as.integer(Price)~ Year + UsedOrNew + Transmission +   
 Engine + DriveType + FuelType + FuelConsumption   
 + Kilometres  
 + Cylinders + BodyType + Doors + Seats + DollarAustralian +  
 PriceIndex ,family= poisson(link= "sqrt"),v\_train\_i1)  
pred\_val <-predict(mg13\_1, newdata = v\_train\_1, ty="response")  
mg13\_pr1\_val <- mean(abs(as.integer(v\_train\_1$Price) - round(pred\_val,0)))  
mg13\_pr1\_tr <- mean(abs(as.integer(v\_train\_i1$Price) -   
 round(mg13\_1$fitted.values,0)))  
rC13\_1 <- 1- (mg13\_1$deviance/mg13\_1$null.deviance)   
  
mg13\_2 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +  
 DriveType +  
 FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= poisson(link= "sqrt"),v\_train\_i2)  
pred\_val <-predict(mg13\_2, newdata = v\_train\_2, ty="response")  
mg13\_pr2\_val <- mean(abs(as.integer(v\_train\_2$Price) - round(pred\_val,0)))  
mg13\_pr2\_tr <- mean(abs(as.integer(v\_train\_i2$Price) -   
 round(mg13\_2$fitted.values,0)))  
rC13\_2 <- 1- (mg13\_2$deviance/mg13\_2$null.deviance)   
  
mg13\_3 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
 DriveType +   
 FuelType + FuelConsumption + Kilometres + Cylinders +  
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= poisson(link= "sqrt"),v\_train\_i3)  
pred\_val <-predict(mg13\_3, newdata = v\_train\_3, ty="response")  
mg13\_pr3\_val <- mean(abs(as.integer(v\_train\_3$Price) - round(pred\_val,0)))  
mg13\_pr3\_tr <- mean(abs(as.integer(v\_train\_i3$Price) -   
 round(mg13\_3$fitted.values,0)))  
rC13\_3 <- 1- (mg13\_3$deviance/mg13\_3$null.deviance)   
  
mg13\_4 <- glm(Price~ Year + UsedOrNew + Transmission + Engine +   
 DriveType +   
 FuelType + FuelConsumption + Kilometres + Cylinders +   
 BodyType + Doors + Seats + DollarAustralian + PriceIndex,  
 family= poisson(link= "sqrt"),v\_train\_i4)  
pred\_val <-predict(mg13\_4, newdata = v\_train\_4, ty="response")  
mg13\_pr4\_val <- mean(abs(as.integer(v\_train\_4$Price) - round(pred\_val,0)))  
mg13\_pr4\_tr <- mean(abs(as.integer(v\_train\_i4$Price) -   
 round(mg13\_4$fitted.values,0)))  
rC13\_4 <- 1- (mg13\_4$deviance/mg13\_4$null.deviance)   
  
#MAE\_v  
mg13\_pr\_val <- mean(c(mg13\_pr1\_val, mg13\_pr2\_val, mg13\_pr3\_val,  
 mg13\_pr4\_val))  
#MAE\_train  
mg13\_pr\_tr <- mean(c(mg13\_pr1\_tr, mg13\_pr2\_tr, mg13\_pr3\_tr,   
 mg13\_pr4\_tr))  
#mean(pseudo R^2)  
rC13 <- mean(c(rC13\_1, rC13\_2, rC13\_3, rC13\_4))  
c(mg13\_pr\_val, mg13\_pr\_tr, rC13)

## [1] 7627.6240238 7613.9599924 0.7367348

#Modelo lineal final obtenido: poisson con familia log  
mg\_full <-glm(as.integer(Price)~ Year + UsedOrNew + Transmission +  
 Engine + DriveType + FuelType + FuelConsumption   
 + Kilometres   
 + Cylinders + BodyType + Doors + Seats +   
 DollarAustralian +   
 PriceIndex ,family= poisson(link= "log"),v\_train)  
  
  
# Optimización del modelo y validación  
  
# Selección de parámetros. No se elimina ninguno.  
library(car)

## Loading required package: carData

library(MASS)  
stepAIC(mg\_full)

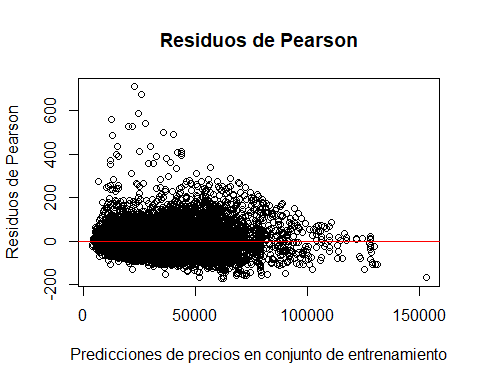
## Start: AIC=39951242  
## as.integer(Price) ~ Year + UsedOrNew + Transmission + Engine +   
## DriveType + FuelType + FuelConsumption + Kilometres + Cylinders +   
## BodyType + Doors + Seats + DollarAustralian + PriceIndex  
##   
## Df Deviance AIC  
## <none> 39778481 39951242  
## - UsedOrNew 1 39781812 39954572  
## - Doors 1 39790904 39963664  
## - Transmission 1 39793981 39966741  
## - PriceIndex 1 39920238 40092998  
## - DollarAustralian 1 39946437 40119197  
## - Seats 1 39976552 40149312  
## - FuelConsumption 1 40742429 40915189  
## - BodyType 3 41187735 41360491  
## - Engine 1 41216520 41389280  
## - Cylinders 1 41390454 41563214  
## - FuelType 2 43761688 43934446  
## - Kilometres 1 44259910 44432670  
## - Year 1 44887799 45060559  
## - DriveType 3 45044310 45217066

##   
## Call: glm(formula = as.integer(Price) ~ Year + UsedOrNew + Transmission +   
## Engine + DriveType + FuelType + FuelConsumption + Kilometres +   
## Cylinders + BodyType + Doors + Seats + DollarAustralian +   
## PriceIndex, family = poisson(link = "log"), data = v\_train)  
##   
## Coefficients:  
## (Intercept) Year UsedOrNewUSED   
## -1.153e+02 6.183e-02 -1.068e-02   
## TransmissionManual Engine DriveTypeFront   
## -2.133e-02 1.415e-01 -2.930e-01   
## DriveTypeOther DriveTypeRear FuelTypeOther   
## 5.729e-03 -1.030e-01 1.360e-02   
## FuelTypeUnleaded FuelConsumption Kilometres   
## -2.105e-01 -3.114e-02 -2.373e-06   
## Cylinders BodyTypeOther BodyTypeSedan   
## 1.095e-01 2.358e-01 1.343e-01   
## BodyTypeSUV/Ute/Tray Doors Seats   
## 1.012e-01 -1.085e-02 2.400e-02   
## DollarAustralian PriceIndex   
## 2.180e-01 2.679e-03   
##   
## Degrees of Freedom: 14300 Total (i.e. Null); 14281 Residual  
## Null Deviance: 165300000   
## Residual Deviance: 39780000 AIC: 39950000

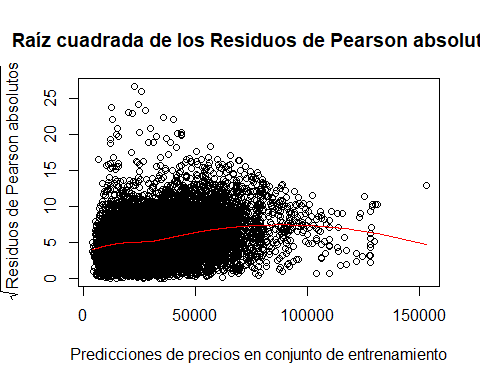
summary(v)

## Year UsedOrNew Transmission Engine   
## Min. :2000 Length:15891 Length:15891 Min. :1.000   
## 1st Qu.:2013 Class :character Class :character 1st Qu.:2.000   
## Median :2017 Mode :character Mode :character Median :2.200   
## Mean :2016 Mean :2.341   
## 3rd Qu.:2020 3rd Qu.:2.500   
## Max. :2023 Max. :6.000   
## DriveType FuelType FuelConsumption Kilometres   
## Length:15891 Length:15891 Min. : 0.000 Min. : 1   
## Class :character Class :character 1st Qu.: 6.700 1st Qu.: 43866   
## Mode :character Mode :character Median : 7.600 Median : 85952   
## Mean : 7.653 Mean : 98446   
## 3rd Qu.: 8.500 3rd Qu.:143000   
## Max. :15.000 Max. :350000   
## Cylinders BodyType Doors Seats   
## Min. :3.00 Length:15891 Min. :2.000 Min. :2.000   
## 1st Qu.:4.00 Class :character 1st Qu.:4.000 1st Qu.:5.000   
## Median :4.00 Mode :character Median :4.000 Median :5.000   
## Mean :4.34 Mean :4.026 Mean :5.099   
## 3rd Qu.:4.00 3rd Qu.:4.000 3rd Qu.:5.000   
## Max. :8.00 Max. :5.000 Max. :8.000   
## Price DollarAustralian PriceIndex   
## Min. : 88 Min. :1.241 Min. : 65.79   
## 1st Qu.: 18990 1st Qu.:1.473 1st Qu.:111.14   
## Median : 28990 Median :1.517 Median :120.42   
## Mean : 33716 Mean :1.534 Mean :117.44   
## 3rd Qu.: 42422 3rd Qu.:1.611 3rd Qu.:127.90   
## Max. :139990 Max. :1.773 Max. :135.76

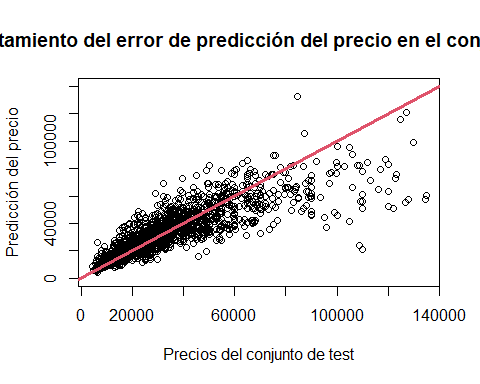
# Gráficos de validación y coeficiente de determinación.   
rC <- 1- (mg\_full$deviance/mg\_full$null.deviance)   
par(mfrow = c(1, 1))  
plot(mg\_full$fitted.values, residuals(mg\_full, type = "pearson"),  
 xlab = "Predicciones de precios en conjunto de entrenamiento",  
 ylab = "Residuos de Pearson",  
 main = "Residuos de Pearson")  
abline(h = 0, col = "red") # Línea horizontal en y = 0



d\_aux <- data.frame(x = mg\_full$fitted.values,   
 y = predict(loess(sqrt(abs(residuals(mg\_full,   
 type = "pearson")))   
 ~ mg\_full$fitted.values)))  
d\_aux <- d\_aux[order(d\_aux$x), ]  
plot(mg\_full$fitted.values, sqrt(abs(residuals(mg\_full,   
 type = "pearson"))),  
 xlab = "Predicciones de precios en conjunto de entrenamiento",  
 ylab = expression(sqrt("Residuos de Pearson absolutos")),  
 main = "Raíz cuadrada de los Residuos de Pearson absolutos")  
# Línea horizontal en y = 0  
#lines(x, lm(y~poly(x,3))$fitted.values, col = "red")  
lines(d\_aux$x, d\_aux$y, col = "red")



# Capacidad predictiva respecto al conjunto de test.  
pred\_test <-round(predict(mg\_full, newdata = v\_test, ty="response"),0)  
err\_test <- mean(abs(as.integer(v\_test$Price) - round(pred\_test,0)))  
  
  
# Interpretación y extracción de conocimiento  
  
# Distribución de error en conjunto de test  
plot(v\_test$Price, pred\_test, xlab = 'Precios del conjunto de test',   
ylab = 'Predicción del precio', main =   
'Comportamiento del error de predicción del precio en el conjunto de test',  
 ylim = c(0,140000))  
abline(a=0, b=1, col=2, lw=3)



aux2 <- v\_test[v\_test$Price > 50000,]  
pred\_test\_st <-round(predict(mg\_full, newdata = aux2, ty="response"),0)   
err\_test\_st <- mean(abs(as.integer(aux2$Price) - round(pred\_test\_st,0)))  
  
# Efectos de variables explicativas  
exp(mg\_full$coefficients)

## (Intercept) Year UsedOrNewUSED   
## 8.220319e-51 1.063782e+00 9.893776e-01   
## TransmissionManual Engine DriveTypeFront   
## 9.788958e-01 1.151982e+00 7.460424e-01   
## DriveTypeOther DriveTypeRear FuelTypeOther   
## 1.005745e+00 9.021227e-01 1.013692e+00   
## FuelTypeUnleaded FuelConsumption Kilometres   
## 8.101853e-01 9.693422e-01 9.999976e-01   
## Cylinders BodyTypeOther BodyTypeSedan   
## 1.115687e+00 1.265864e+00 1.143764e+00   
## BodyTypeSUV/Ute/Tray Doors Seats   
## 1.106485e+00 9.892131e-01 1.024289e+00   
## DollarAustralian PriceIndex   
## 1.243562e+00 1.002683e+00

# Ejemplos de coches más baratos y más caros posibles.  
vb <-data.frame(Year = 2000, UsedOrNew = 'USED', Transmission = 'Manual',   
 Engine = 1, FuelType = 'Unleaded',   
 DriveType = 'Front', FuelConsumption = 15,   
 Kilometres = 350000, Cylinders = 3,   
 BodyType = 'SUV/Ute/Tray', Doors = 5, Seats = 2,   
 DollarAustralian = 1.241, PriceIndex = 65.79)  
  
vc <-data.frame(Year = 2023, UsedOrNew = 'NOT USED',   
 Transmission = 'Automatic', Engine = 6,   
 FuelType = 'Other',   
 DriveType = 'Other', FuelConsumption = 0, Kilometres = 1,   
 Cylinders = 8, BodyType = 'Other', Doors = 2, Seats = 8,   
 DollarAustralian = 1.773, PriceIndex = 135.76)  
  
# Precio medio de coche más caro e interval de confianza.  
predict(mg\_full, newdata = vc,ty="response")

## 1   
## 314464.3

vc\_l <- predict(mg\_full, vc, ty="link")  
exp(vc\_l + ((predict(mg\_full,newdata = vc, ty = "link",se.fit=T)$se.fit)\*   
 qnorm(0.975)))

## 1   
## 314802.6

exp(vc\_l + ((predict(mg\_full,newdata = vc, ty = "link",se.fit=T)$se.fit)\*  
 qnorm(0.025)))

## 1   
## 314126.3

# Precio medio de coche más barato e interval de confianza.  
predict(mg\_full, newdata = vb,ty="response")

## 1   
## 1835.126

vb\_l <- predict(mg\_full, vb, ty="link")  
exp(vb\_l + ((predict(mg\_full,newdata = vb, ty = "link",se.fit=T)$se.fit)\*   
 qnorm(0.975)))

## 1   
## 1836.825

exp(vb\_l + ((predict(mg\_full,newdata = vb, ty = "link",se.fit=T)$se.fit)\*   
 qnorm(0.025)))

## 1   
## 1833.429