Lab 09

## setup

setwd("C:/Users/22700/Desktop")  
library(data.table)  
library(GGally)

## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

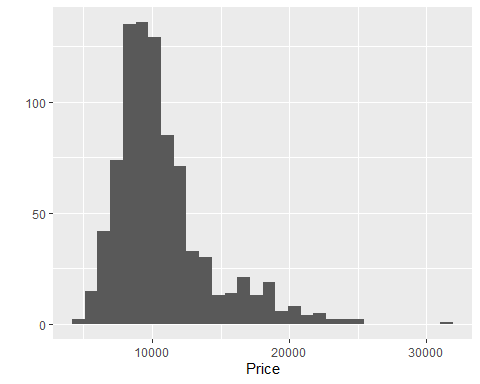
load("car.test.RData")  
load("car.train.Rdata")

## 1. Use the dataset carTrain.RData to build a model explaining the price of used cars.

#1. a) Explore the dataset and obtain descriptive statistics.

dt.car.train <- data.table(car.train)  
rm(car.train)  
dt.car.test <- data.table(car.test)  
rm(car.test)  
qplot( data = dt.car.train  
, x = Price  
, geom = "histogram")

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

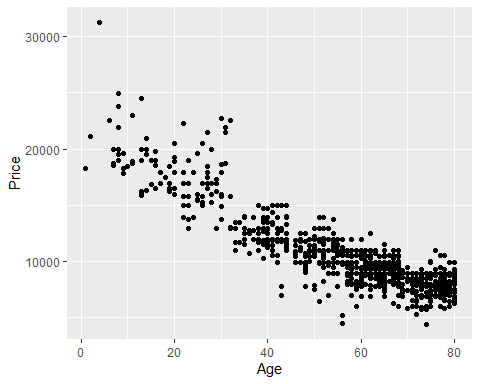
 # For linear regression, you ideally want to make sure that your target variable is normally distributed (or nearly normal). You also want the relationship between the dep. var. and indep. vars. to be as linear as possible. Constant variance of the dep. var. is also good.

cor(dt.car.train[, list(Price, Age, KM, HP, CC, Doors, Weight)])

## Price Age KM HP CC Doors  
## Price 1.0000000 -0.8825304 -0.571087610 0.31044297 0.13644094 0.22177598  
## Age -0.8825304 1.0000000 0.502809576 -0.14722197 -0.11441204 -0.19251504  
## KM -0.5710876 0.5028096 1.000000000 -0.33284208 0.32149979 -0.04203052  
## HP 0.3104430 -0.1472220 -0.332842083 1.00000000 0.06462772 0.10127366  
## CC 0.1364409 -0.1144120 0.321499789 0.06462772 1.00000000 0.14501267  
## Doors 0.2217760 -0.1925150 -0.042030523 0.10127366 0.14501267 1.00000000  
## Weight 0.5565002 -0.4627714 -0.006362284 0.09256393 0.65806566 0.33138029  
## Weight  
## Price 0.556500240  
## Age -0.462771440  
## KM -0.006362284  
## HP 0.092563932  
## CC 0.658065655  
## Doors 0.331380291  
## Weight 1.000000000

# Based on the correlation table, Age seems to be the most important predictor of Price.

qplot( data = dt.car.train  
, x = Age  
, y = Price  
, geom = "point")



## Alternatively, you can use ggpairs to look at the relationships between the variables (note that ggpairs can be somewhat time consuming).

set.seed(201410)  
ix <- sample(1:nrow(dt.car.train), size = nrow(dt.car.train) \* 0.2)  
#ggpairs(dt.car.train[ix,])

## 1.b) How good is Age at predicting Price?

out0 <- lm( Price ~ Age , data = dt.car.train)  
stargazer(out0, type = "text")

##   
## ===============================================  
## Dependent variable:   
## ---------------------------  
## Price   
## -----------------------------------------------  
## Age -167.361\*\*\*   
## (3.041)   
##   
## Constant 20,056.420\*\*\*   
## (179.011)   
##   
## -----------------------------------------------  
## Observations 862   
## R2 0.779   
## Adjusted R2 0.779   
## Residual Std. Error 1,653.344 (df = 860)   
## F Statistic 3,028.937\*\*\* (df = 1; 860)   
## ===============================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

As a curiosity (you do not need to know this), there is an automatic way to chose the model using the Akaike information criterion (AIC): “The AIC is a measure of the relative quality of a statistical model for a given set of data. As such, AIC provides a means for model selection. **AIC deals with the trade-off between the goodness of fit of the model and the complexity of the model. It is founded on information theory: it offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. AIC does not provide a test of a model in the sense of testing a null hypothesis; i.e. AIC can tell nothing about the quality of the model in an absolute sense. If all the candidate models fit poorly, AIC will not give any warning of that.” (Source: Wikipedia)**

To chose the model following AIC you use the function ‘step’: start by running a model with all the variables in the data set and then use the step function to select which variables should be included in the model. Note that this function does not create new variables, it only tests models with the variables that already exist in your dataset. Thus, you need to create the squared or logged versions of the variables yourself in case you want them to be considered by the step function.

## 1.c) Use the function step to improve your prediction model.

out1 <- lm(Price ~ . , data = dt.car.train)  
summary(step(out1))

## Start: AIC=12328.75  
## Price ~ Age + KM + FuelType + HP + MetColor + Automatic + CC +   
## Doors + Weight  
##   
## Df Sum of Sq RSS AIC  
## - Doors 1 179666 1367641272 12327  
## - MetColor 1 474439 1367936045 12327  
## <none> 1367461607 12329  
## - Automatic 1 8691751 1376153358 12332  
## - FuelType 2 44312263 1411773870 12352  
## - CC 1 64789102 1432250709 12367  
## - HP 1 116727742 1484189348 12397  
## - KM 1 139399741 1506861348 12410  
## - Weight 1 216426919 1583888525 12453  
## - Age 1 2310948919 3678410526 13180  
##   
## Step: AIC=12326.86  
## Price ~ Age + KM + FuelType + HP + MetColor + Automatic + CC +   
## Weight  
##   
## Df Sum of Sq RSS AIC  
## - MetColor 1 501801 1368143073 12325  
## <none> 1367641272 12327  
## - Automatic 1 8512151 1376153424 12330  
## - FuelType 2 45244655 1412885928 12351  
## - CC 1 65803225 1433444498 12365  
## - HP 1 120648969 1488290242 12398  
## - KM 1 139342595 1506983868 12408  
## - Weight 1 255983844 1623625116 12473  
## - Age 1 2315709821 3683351093 13179  
##   
## Step: AIC=12325.17  
## Price ~ Age + KM + FuelType + HP + Automatic + CC + Weight  
##   
## Df Sum of Sq RSS AIC  
## <none> 1368143073 12325  
## - Automatic 1 8278648 1376421721 12328  
## - FuelType 2 44943988 1413087060 12349  
## - CC 1 65368607 1433511679 12363  
## - HP 1 120420014 1488563087 12396  
## - KM 1 140102634 1508245706 12407  
## - Weight 1 256991562 1625134635 12472  
## - Age 1 2318274423 3686417496 13178

##   
## Call:  
## lm(formula = Price ~ Age + KM + FuelType + HP + Automatic + CC +   
## Weight, data = dt.car.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9774.4 -740.3 14.4 715.0 6571.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.097e+03 1.601e+03 -1.310 0.1904   
## Age -1.233e+02 3.243e+00 -38.018 < 2e-16 \*\*\*  
## KM -1.516e-02 1.622e-03 -9.346 < 2e-16 \*\*\*  
## FuelTypeDiesel 3.563e+03 6.819e+02 5.224 2.20e-07 \*\*\*  
## FuelTypePetrol 1.154e+03 4.592e+02 2.513 0.0122 \*   
## HP 6.324e+01 7.299e+00 8.665 < 2e-16 \*\*\*  
## Automatic 4.294e+02 1.890e+02 2.272 0.0233 \*   
## CC -4.380e+00 6.861e-01 -6.384 2.83e-10 \*\*\*  
## Weight 1.843e+01 1.456e+00 12.658 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1266 on 853 degrees of freedom  
## Multiple R-squared: 0.8713, Adjusted R-squared: 0.8701   
## F-statistic: 721.9 on 8 and 853 DF, p-value: < 2.2e-16

## 2. Did you use all the variables in the dataset to build your model? Why?

out2 <- lm(Price ~ Age + KM + FuelType + HP + Automatic + CC + Weight , data = dt.car.train)

## I did not use MetColor and Doors. When added to the model, these variables are not individually significant and they do not raise the model’s R2.

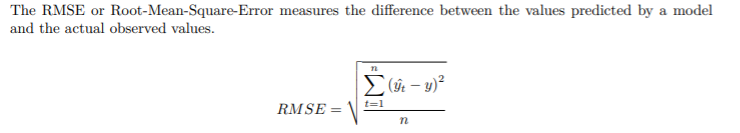
## 3. Use your model to predict used car prices in the datset carTest.RData.

dt.car.test <- dt.car.test[, yhat:=predict(out2, newdata=dt.car.test)]  
head(dt.car.test)

## Price Age KM FuelType HP MetColor Automatic CC Doors Weight yhat  
## 1: 11290 49 80320 Petrol 110 1 1 1600 3 1070 11898.135  
## 2: 15950 19 51884 Petrol 97 1 0 1400 3 1100 16205.945  
## 3: 8500 80 100458 Petrol 110 0 0 1600 5 1085 7617.351  
## 4: 8900 67 54847 Petrol 110 0 0 1600 3 1050 9266.716  
## 5: 15950 28 29206 Petrol 97 1 0 1400 5 1110 15624.302  
## 6: 15950 30 67660 Petrol 110 1 0 1600 3 1105 14648.714

## 4. Use the RMSE to compare the performance of your model in carTrain.RData and carTest.RData.

library(Metrics)



#TRAIN

dt.car.train <- dt.car.train[, yhat:=predict(out2, newdata=dt.car.train)]  
dt.car.train <- dt.car.train[, uhat:=yhat-Price]  
dt.car.train <- dt.car.train[, uhat2:=uhat^2]  
n.train <- nrow(dt.car.train)  
rmse.train <- sqrt(sum(dt.car.train$uhat2)/n.train)  
rmse.train

## [1] 1259.831

#TEST

dt.car.test <- dt.car.test[, yhat:=predict(out2, newdata=dt.car.test)]  
dt.car.test <- dt.car.test[, uhat:=yhat-Price]  
dt.car.test <- dt.car.test[, uhat2:=uhat^2]  
n.test <- nrow(dt.car.test)  
rmse.test <- sqrt(sum(dt.car.test$uhat2)/n.test)  
rmse.test

## [1] 1389.831