Unit 3 Homework - Housing regression with AES data

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

# load packages

packages <- c("AER", "ggplot2", "PerformanceAnalytics", "plyr")  
  
sapply(packages, library, character.only = TRUE)

## Loading required package: car

## Loading required package: lmtest

## Loading required package: zoo

##   
## Attaching package: 'zoo'

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

## Loading required package: sandwich

## Loading required package: survival

## Loading required package: xts

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

# Read in the data

data("HousePrices")  
  
df<- HousePrices

# Quick EDA

summary(df)

## price lotsize bedrooms bathrooms   
## Min. : 25000 Min. : 1650 Min. :1.000 Min. :1.000   
## 1st Qu.: 49125 1st Qu.: 3600 1st Qu.:2.000 1st Qu.:1.000   
## Median : 62000 Median : 4600 Median :3.000 Median :1.000   
## Mean : 68122 Mean : 5150 Mean :2.965 Mean :1.286   
## 3rd Qu.: 82000 3rd Qu.: 6360 3rd Qu.:3.000 3rd Qu.:2.000   
## Max. :190000 Max. :16200 Max. :6.000 Max. :4.000   
## stories driveway recreation fullbase gasheat aircon   
## Min. :1.000 no : 77 no :449 no :355 no :521 no :373   
## 1st Qu.:1.000 yes:469 yes: 97 yes:191 yes: 25 yes:173   
## Median :2.000   
## Mean :1.808   
## 3rd Qu.:2.000   
## Max. :4.000   
## garage prefer   
## Min. :0.0000 no :418   
## 1st Qu.:0.0000 yes:128   
## Median :0.0000   
## Mean :0.6923   
## 3rd Qu.:1.0000   
## Max. :3.0000

str(df)

## 'data.frame': 546 obs. of 12 variables:  
## $ price : num 42000 38500 49500 60500 61000 66000 66000 69000 83800 88500 ...  
## $ lotsize : num 5850 4000 3060 6650 6360 4160 3880 4160 4800 5500 ...  
## $ bedrooms : num 3 2 3 3 2 3 3 3 3 3 ...  
## $ bathrooms : num 1 1 1 1 1 1 2 1 1 2 ...  
## $ stories : num 2 1 1 2 1 1 2 3 1 4 ...  
## $ driveway : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...  
## $ recreation: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 2 2 ...  
## $ fullbase : Factor w/ 2 levels "no","yes": 2 1 1 1 1 2 2 1 2 1 ...  
## $ gasheat : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ aircon : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 1 2 ...  
## $ garage : num 1 0 0 0 0 0 2 0 0 1 ...  
## $ prefer : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

head(df)

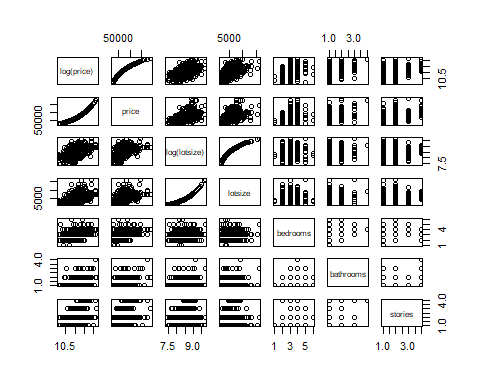
## price lotsize bedrooms bathrooms stories driveway recreation fullbase  
## 1 42000 5850 3 1 2 yes no yes  
## 2 38500 4000 2 1 1 yes no no  
## 3 49500 3060 3 1 1 yes no no  
## 4 60500 6650 3 1 2 yes yes no  
## 5 61000 6360 2 1 1 yes no no  
## 6 66000 4160 3 1 1 yes yes yes  
## gasheat aircon garage prefer  
## 1 no no 1 no  
## 2 no no 0 no  
## 3 no no 0 no  
## 4 no no 0 no  
## 5 no no 0 no  
## 6 no yes 0 no

### Looks like we have a few factors that are 2 levels and not boolean. Although R will fix this automatically in the regression, lets fix that:

df$driveway <- as.numeric(df$driveway =='yes')  
df$recreation<- as.numeric(df$recreation =='yes')  
df$fullbase <- as.numeric(df$fullbase =='yes')  
df$gasheat <- as.numeric(df$gasheat =='yes')  
df$aircon <- as.numeric(df$aircon == 'yes')  
df$prefer <- as.numeric(df$prefer == 'yes')

### let's look at some breakdowns of the data:

df\_names <- names(df)  
  
#scatterplotMatrix(~price + lotsize + bedrooms + bathrooms , data = df)  
pairs(~log(price) +price + log(lotsize) + lotsize + bedrooms + bathrooms +stories , data = df)



### correlations:

# cor(df)  
  
  
# log-base  
cor(log(df$price), df$lotsize)

## [1] 0.5429071

cor(log(df$price), df$bedrooms)

## [1] 0.3698846

cor(log(df$price), df$bathrooms)

## [1] 0.4846417

cor(log(df$price), df$stories)

## [1] 0.4161148

# base-base  
  
cor(df$price, df$lotsize)

## [1] 0.5357957

cor(df$price, df$bedrooms)

## [1] 0.3664474

cor(df$price, df$bathrooms)

## [1] 0.5167193

cor(df$price, df$stories)

## [1] 0.4211902

# log-log  
  
cor(log(df$price), log(df$lotsize))

## [1] 0.5799856

cor(log(df$price), log(df$bedrooms))

## [1] 0.3908739

cor(log(df$price), log(df$bathrooms))

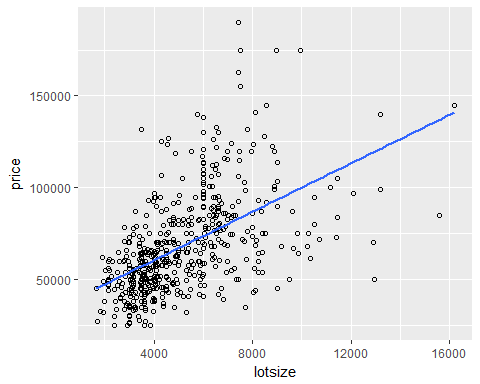
## [1] 0.4893604

cor(log(df$price), log(df$stories))

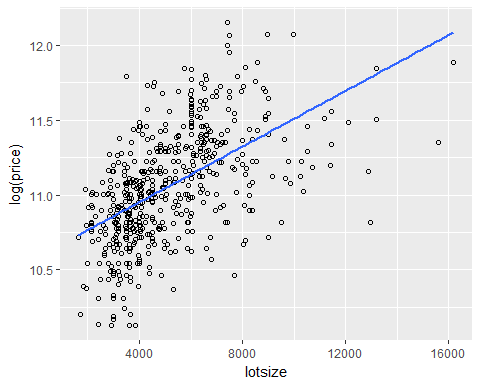
## [1] 0.3833027

### the scatterplot matrix is interesting,let's take a deeper look:

par(mfrow = c(3,1))  
  
# plot price versus lot size  
ggplot(df, aes(y = price, x = lotsize)) + geom\_point(shape = 1) + geom\_smooth(method = lm, se = FALSE)

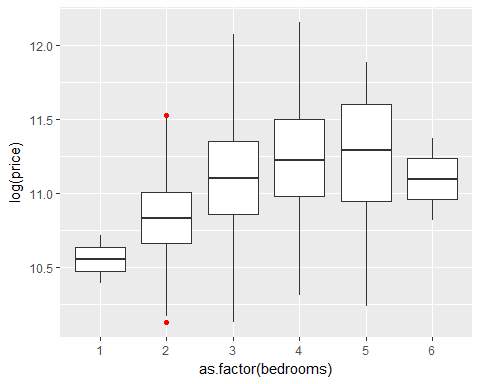


# plots log(price) versus lot size  
  
ggplot(df, aes(y = log(price), x = lotsize)) + geom\_point(shape = 1) + geom\_smooth(method = lm, se = FALSE)



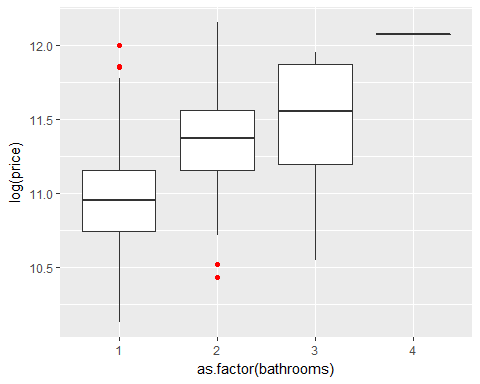
# plots log(price) by bedrooms  
  
ggplot(df, aes(x =as.factor(bedrooms), y = log(price)), color = bedrooms) + geom\_boxplot(outlier.colour = 'red', fun.y = mean)

## Warning: Ignoring unknown parameters: fun.y



# interesting that when 6 bedrooms are hit that the price drops, I bet there are fewer observations here  
  
ggplot(df, aes(x =as.factor(bathrooms), y = log(price)), color = bathrooms) + geom\_boxplot(outlier.colour = 'red', fun.y = mean)

## Warning: Ignoring unknown parameters: fun.y



# must be very few 4 bathroom houses as well

count(df, 'bedrooms')

## bedrooms freq  
## 1 1 2  
## 2 2 136  
## 3 3 301  
## 4 4 95  
## 5 5 10  
## 6 6 2

# there are only 2-6 bedroom houses and 2 - 1 bedrooms houses

count(df, 'bathrooms')

## bathrooms freq  
## 1 1 402  
## 2 2 133  
## 3 3 10  
## 4 4 1

# there is only 1- 4 bathroom house

# Create training and test sets of data:

# doing an 80-20 split for convenience:  
  
# create the holdout indices:  
set.seed(100)  
  
indices <- sample(1:nrow(df), size = nrow(df) \*.2)  
training<- df[-indices,]  
test <- df[indices,]  
  
# check to make sure that it worked (sb 0)  
  
print(nrow(df) - (nrow(training) + nrow(test)))

## [1] 0

# Baseline naive regression:

# the goal is to beat this naive forecast:  
  
  
naive\_lm <- lm(price~ lotsize, data = training)

### Naive summary:

summary(naive\_lm)

##   
## Call:  
## lm(formula = price ~ lotsize, data = training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -69092 -14171 -2416 9925 107282   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.386e+04 2.665e+03 12.71 <2e-16 \*\*\*  
## lotsize 6.585e+00 4.735e-01 13.91 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 21730 on 435 degrees of freedom  
## Multiple R-squared: 0.3078, Adjusted R-squared: 0.3062   
## F-statistic: 193.4 on 1 and 435 DF, p-value: < 2.2e-16

### Naive ANOVA

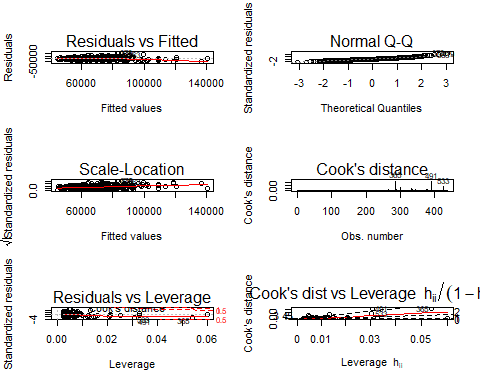
Anova(naive\_lm)

## Anova Table (Type II tests)  
##   
## Response: price  
## Sum Sq Df F value Pr(>F)   
## lotsize 9.1323e+10 1 193.39 < 2.2e-16 \*\*\*  
## Residuals 2.0541e+11 435   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

It does appear that the lot size variable is significant

### Diagnostics:

par(mfrow = c(3,2))  
  
plot(naive\_lm, which = 1:6)



par(mfrow = c(1,1))

### based on the residuals of this model there are a few things to note:

1.) There appears to be some heteroskedasticity in the residuals, this may be due to sparsity in the data, it may also be because we have not transformed any variables yet  
2.) This also means that the residuals are not normally distributed (potentially based on QQplot)  
3.) WE do seem to have a few highly influential outliers; although maybe not as much of a concern given that we are training on 437 observations

### Naive accuracy:

naive\_pred<- predict(naive\_lm, test)  
  
MAPE <- mean(abs(naive\_pred / test$price -1))  
  
MAPE

## [1] 0.3015798

### Now that we have established a baseline error of 30% to beat, let's try to beat it:

### Model 1: add bathrooms, bedrooms, stories

lm<- lm(log(price)~ lotsize + bathrooms + bedrooms + stories, data = training)  
  
summary(lm)

##   
## Call:  
## lm(formula = log(price) ~ lotsize + bathrooms + bedrooms + stories,   
## data = training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.75701 -0.17070 0.01198 0.16465 0.63276   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.006e+01 5.418e-02 185.685 < 2e-16 \*\*\*  
## lotsize 7.744e-05 5.577e-06 13.886 < 2e-16 \*\*\*  
## bathrooms 1.977e-01 2.748e-02 7.196 2.77e-12 \*\*\*  
## bedrooms 5.210e-02 1.828e-02 2.850 0.00459 \*\*   
## stories 1.061e-01 1.555e-02 6.823 3.03e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2491 on 432 degrees of freedom  
## Multiple R-squared: 0.5378, Adjusted R-squared: 0.5335   
## F-statistic: 125.7 on 4 and 432 DF, p-value: < 2.2e-16

It looks like the adjusted r-squared went up a good bit in this model and that all of the variables appear to be significatnt. We also appear to be getting an overall significant result besed on the overall F-test.

We also need to check for collinearity:

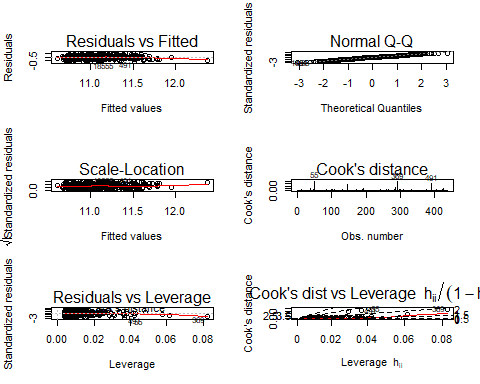
vif(lm)

## lotsize bathrooms bedrooms stories   
## 1.055422 1.264454 1.315752 1.248433

This is actually a good sign. It appears that there is very little collinearity between the variables (this was also evidenced in the previous observations of correlations in the scatterplot matrix)

### Diagnostics:

par(mfrow = c(3,2))  
  
plot(lm, which = 1:6)



par(mfrow = c(1,1))

This looks alot better, the redisuals appear to be normally distributed and they do appear to be homoskedastic (despite the line looking odd - this is just because of the lack of observations, we still maintain residuals within the range). There do appear to be some leverage, points, but we can address those later if we really need to tunr the model. The Cook's D is still very low overall.

Test the accuracy:

pred\_1 <- predict(lm, test)  
  
MAPE <- mean(abs(pred\_1 / log(test$price) -1))  
  
MAPE

## [1] 0.0206497

This model produces an error rate of 2.0%, pretty impressive improvement with just a few additional variables

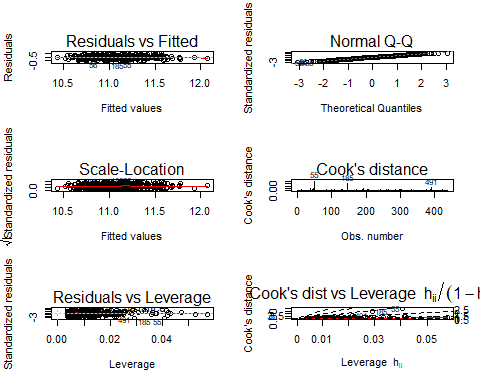
### Model 2 : transform lotsize

training\_2<- training  
training\_2$lotsize <- log(training\_2$lotsize)  
  
test\_2<- training  
test\_2$lotsize <- log(test\_2$lotsize)  
  
  
lm\_2<- lm(log(price)~ lotsize + bathrooms + bedrooms + stories, data = training\_2)  
  
summary(lm\_2)

##   
## Call:  
## lm(formula = log(price) ~ lotsize + bathrooms + bedrooms + stories,   
## data = training\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.74373 -0.15996 0.01066 0.15899 0.59323   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.59919 0.24596 26.830 < 2e-16 \*\*\*  
## lotsize 0.45748 0.02963 15.439 < 2e-16 \*\*\*  
## bathrooms 0.18959 0.02657 7.136 4.08e-12 \*\*\*  
## bedrooms 0.05418 0.01763 3.073 0.00225 \*\*   
## stories 0.10067 0.01501 6.705 6.30e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2405 on 432 degrees of freedom  
## Multiple R-squared: 0.5692, Adjusted R-squared: 0.5652   
## F-statistic: 142.7 on 4 and 432 DF, p-value: < 2.2e-16

The adj r-squared went up slightly with this model, which could be a good sign given that we did not add any additional variables

par(mfrow = c(3,2))  
  
plot(lm\_2, which = 1:6)



par(mfrow = c(1,1))

This model looks even better

Test the accuracy:

pred\_2 <- predict(lm\_2, test\_2)  
  
MAPE <- mean(abs(pred\_2 / log(test\_2$price) -1))  
  
MAPE

## [1] 0.01697214

Ok, a solid improvement in accuracy of about 40 basis points

### Model 3 : add the log of the remaining variables

training\_3<- training  
training\_3$lotsize <- log(training\_3$lotsize)  
training\_3$bathrooms <- log(training\_3$bathrooms)  
training\_3$bedrooms <- log(training\_3$bedrooms)  
training\_3$stories <- log(training\_3$stories)  
  
test\_3<- test  
test\_3$lotsize <- log(test\_3$lotsize)  
test\_3$bathrooms <- log(test\_3$bathrooms)  
test\_3$bedrooms <- log(test\_3$bedrooms)  
test\_3$stories <- log(test\_3$stories)  
  
  
  
lm\_3<- lm(log(price)~ lotsize + bathrooms + bedrooms + stories, data = training\_3)  
  
summary(lm\_3)

##   
## Call:  
## lm(formula = log(price) ~ lotsize + bathrooms + bedrooms + stories,   
## data = training\_3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.73590 -0.16845 0.02181 0.15794 0.59565   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.80090 0.25013 27.190 < 2e-16 \*\*\*  
## lotsize 0.46514 0.02969 15.665 < 2e-16 \*\*\*  
## bathrooms 0.30301 0.03994 7.586 2.04e-13 \*\*\*  
## bedrooms 0.16861 0.05442 3.098 0.00207 \*\*   
## stories 0.17112 0.02996 5.712 2.08e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2403 on 432 degrees of freedom  
## Multiple R-squared: 0.5701, Adjusted R-squared: 0.5661   
## F-statistic: 143.2 on 4 and 432 DF, p-value: < 2.2e-16

There is very little improvmement in the asj r^2 here. This probably means that we may not be adding that much more explaianbility to the model. We also notice that the coefficients for bedrooms and stories are very small for each percentage change in the respective variable

pred\_3 <- predict(lm\_3, test\_3)  
  
MAPE <- mean(abs(pred\_3 / log(test\_3$price) -1))  
  
MAPE

## [1] 0.0204332

This model was only slightly better than the original (2% error). However, the previous model actually beat this model's accuracy score.

From an accuracy point of view, this is proably the best model, we can look at the ANOVAs as well to get a better understanding of the model's abilities to capture differences.

### ANOVA tests:

#### Naive versus lm\_1:

anova( lm\_2, lm\_3)

## Analysis of Variance Table  
##   
## Model 1: log(price) ~ lotsize + bathrooms + bedrooms + stories  
## Model 2: log(price) ~ lotsize + bathrooms + bedrooms + stories  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 432 24.992   
## 2 432 24.939 0 0.052916

The difference between the 2 models is very small and therefore, not signficant. In this case, we would choose lm\_2

### Conclusion

LM2 produced the highest accuracy and was not significantly different from the model with all variables logged. Therefore, we would select this model as it is simpler and easier to explain.

While we didn't necessarily evaluate the models all solely based on the the ANOVA approach, we did take more of a data mining approach of looking at accuracy, which is what tends to count in the end assuming all assumptions were met. In this case, we met all of the regression assumptions nad produced a model that was more accurate. Going forward, we might consider trying additional regressions based on machine learning classification, such as Support vector regression, artificial neural networks, or boosting