linear_regression_sample.R

nicol

2020-04-16

```
### Linear Regression ### --------------------------------------------------------------------------------

# By: Nicole Davila
# Date: 2020-05-06

### Import required libraries
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.1

## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(dplyr)
```

```
##
## 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
```

```
library(tidyr)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(lm.beta)

### Import data
houses = read.csv("C:/Users/nicol/OneDrive/Public/Email attachments/Documents/R Sample Work/houses.csv")

### Let's the data into a train and test sample such that 70% of the data is in the train sample and pa
```

```r
# Set seed
set.seed(1031)
# Split data
split=createDataPartition(y=houses$price,p=0.7,list=F,groups=100)
train=houses[split,]
test=houses[-split,]
# Check average house price in each sample
mean(train$price)
```

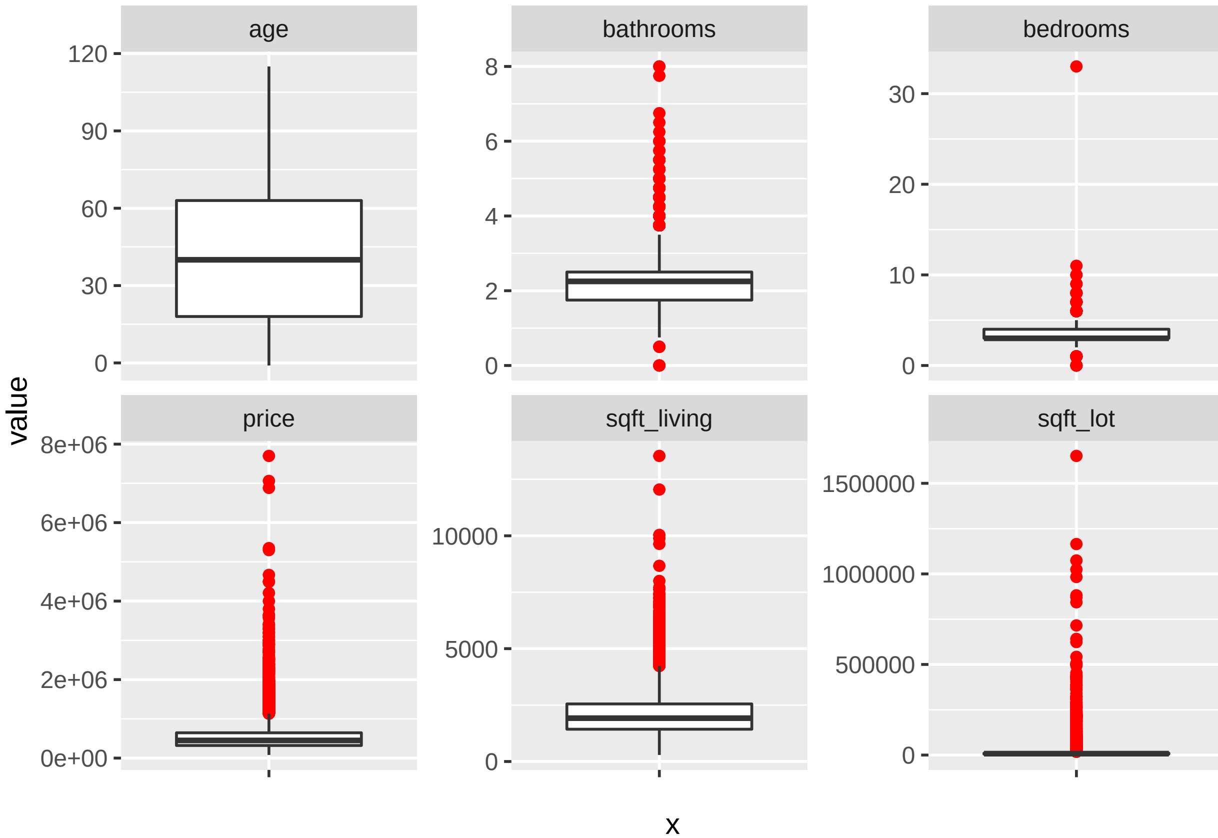
```
## [1] 540165.7
```

```r
# 540674.2
mean(test$price)
```

```
## [1] 539905.5
```

```r
# 538707.6

### Let's do some data exploration on the train sample to better understand the structure and nature of
#   values.
train %>%
  select(id,price:sqft_lot,age)%>%
  gather(key=numericVariable,value=value,price:age)%>%
  ggplot(aes(x='',y=value))+
  geom_boxplot(outlier.color = 'red')+
  facet_wrap(~numericVariable,scales='free_y')
```



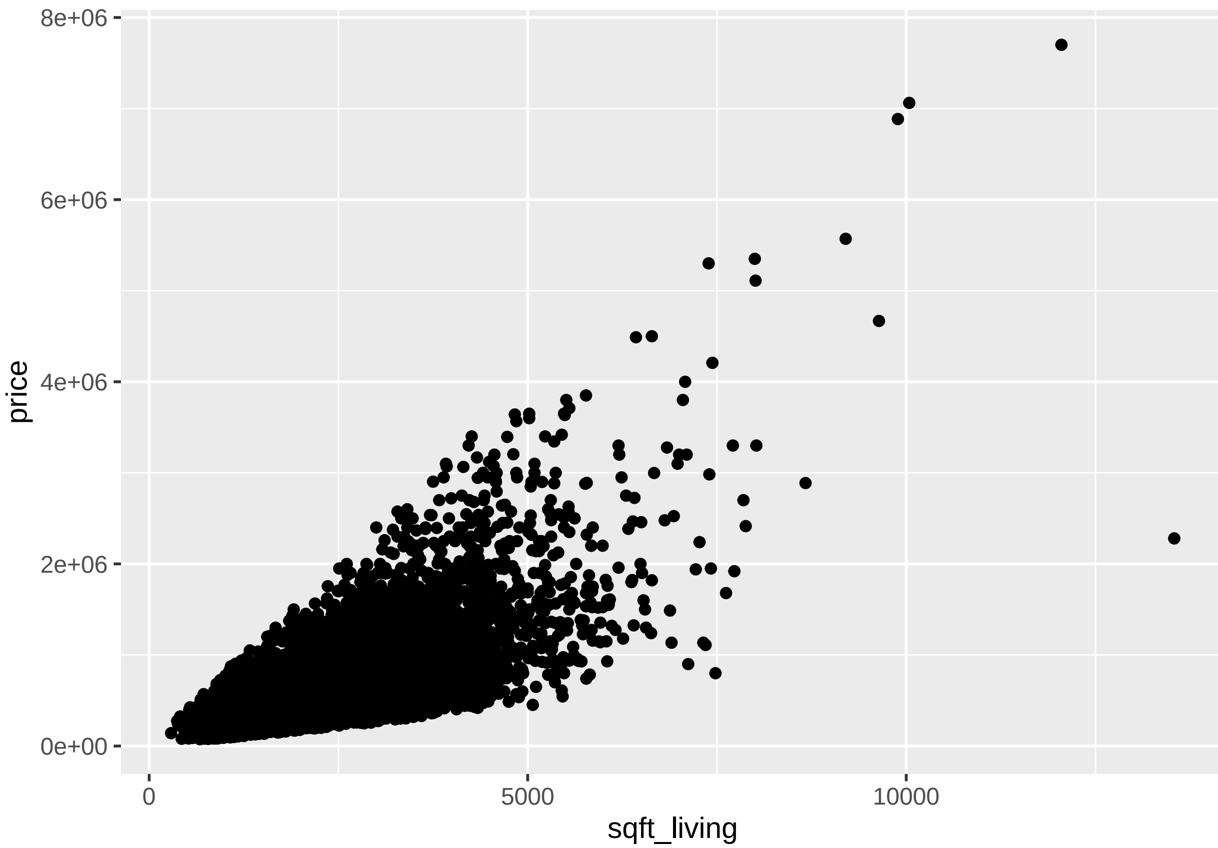

```
# We can see that there are outliers for bathrooms, bedrooms, price, sqft_living, and sqft_lot. Let's i

### Let's see what the living area (sqft_living) for the house with the most bedrooms is.
data.table(train)[bedrooms==max(bedrooms),"sqft_living"]
```

```
##    sqft_living
## 1:        1620
```

```
# 1620

### It is expected that larger houses cost more, but let's onstruct a scatterplot to examine the relati
#   and price, placing sqft_living on the horizontal axis and price on the vertical axis. This will all
#   this hypothesis.
ggplot(data=houses,aes(x=sqft_living, y=price))+
  geom_point()
```




```
# We see the dots going ottom-left to top-right confirming our hypothesis.

### Now let's take a look at the correlation between sqft_living and price?
cor(houses$sqft_living,houses$price)
```

```
## [1] 0.7020351
```

```
# A correlation of 0.7020351, which is relatively close to 1, indicates there is in fact a positive rel
# variables. This aligns with wht we saw in the scatterplot.

### Now, let's construct a simple regression to predict house price from area (sqft_living) using the t
#   and examine how well the model is predicting price by calculating the p-value for the F-statistic.
model1 = lm(price ~ sqft_living, data=train)
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ sqft_living, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1491759  -146386   -24131   106578  4348558
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -47764.278   5250.938  -9.096   <2e-16 ***
## sqft_living    282.092      2.305 122.381   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 261100 on 15170 degrees of freedom
## Multiple R-squared:  0.4968, Adjusted R-squared:  0.4968
## F-statistic: 1.498e+04 on 1 and 15170 DF,  p-value: < 2.2e-16
```

```
# Since we get a p-value of < 2.2e-16, we can say with a good degree of confidence that our model is pe

### Let's calculate the R2 for model1
pred1 = predict(model1)
sse1 = sum((pred1 - train$price)^2)
sst1 = sum((mean(train$price)-train$price)^2)
model1_r2 = 1 - sse1/sst1; model1_r2
```

```
## [1] 0.4967993
```

```
# Since we got an R2 of 0.4985522, we can say that our model expalins about 50% of the variablity of th
# mean. Our model does not fit the data too well.

### Let's see what the rmse for model1 is, since this is an absolute measure of fir, whereas R2 is a re
rmse1 = sqrt(mean((pred1-train$price)^2))
rmse1
```

```
## [1] 261068.9
```

```
# 263932.6
# Note: RMSE can be better interpreted in relation to other models using the same data. Below we will c
#       we will be able to better interpret this measure.

### Since this model is built on sample data, it is important to see if the coefficient estimates are n
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ sqft_living, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1491759  -146386   -24131   106578  4348558
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -47764.278   5250.938  -9.096   <2e-16 ***
## sqft_living    282.092      2.305 122.381   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 261100 on 15170 degrees of freedom
## Multiple R-squared:  0.4968, Adjusted R-squared:  0.4968
## F-statistic: 1.498e+04 on 1 and 15170 DF,  p-value: < 2.2e-16
```

```r
# Based on the model results, indicate your agreement with the following statement we see that the coef
# significantly different from zero.

### Based on this model, on average, what would a 1400 square foot house cost?
predict(model1, newdata = data.frame(sqft_living = 1400))
```

```
##        1
## 347164.3
```

```r
# 346581

### Let's imagine a homeowner were to put in a 200 square foot addition on the house. How much would th
#   up by?
predict(model1, newdata = data.frame(sqft_living = 200)) - predict(model1, newdata = data.frame(sqft_li
```

```
##        1
## 56418.37
```

```r
# 56980.49

### Let's construct another simple regression to predict house price from waterfront. Once again, let's
#   created earlier. Waterfront is a boolean where 1 indicates the house has a view to the waterfront a
model2 = lm(price~waterfront, data = train)
pred2 = predict(model2)
sse2 = sum((pred2 - train$price)^2)
sst2 = sum((mean(train$price)-train$price)^2)
model2_r2 = 1 - sse2/sst2; model2_r2
```

```
## [1] 0.05888983
```

```r
# We get an R2 of 0.07406626 indicating that a waterfront does not really influence the price of a hous

### Let's take a look at the impact of a waterfront view on the expected price. That is, how much more
#   house with a waterfront view compared to one without a waterfront view?
predict(model2, newdata = data.frame(waterfront = 1))- predict(model2, newdata = data.frame(waterfront =
```

```
##       1
## 1103728
```

```r
# 1179766

### We had previously calculated the RMSe for model1. Now let's compare it to the RMSE for model2.
rmse1
```

```
## [1] 261068.9
```

```r
rmse2 = sqrt(mean((pred2 - train$price)^2))
rmse2
```

```
## [1] 357030.1
```

```r
# We see that model1 has an RMSE of 263932.6 which is lower than the RMSE for model2 (358649.4), indica
# Therefore, we could say that the area of a house is a better predictor than a house having a waterfro

### Let's use both the predictors from model1 and model2 to predict price and compare the R2 against th
model3 = lm(price~waterfront+sqft_living, data = train)
pred3 = predict(model3)
sse3 = sum((pred3 - train$price)^2)
sst3 = sum((mean(train$price)-train$price)^2)
model3_r2 = 1 - sse3/sst3
model3_r2
```

```
## [1] 0.5291194
```

```r
model2_r2
```

```
## [1] 0.05888983
```

```r
model1_r2
```

```
## [1] 0.4967993
```

```r
rmse3 = sqrt(mean((pred3 - train$price)^2))
rmse3
```

```
## [1] 252545.7
```

```r
# We see that model3 has an R2 of 0.5375464, which is higher than model1 and model2, indicating better
# RMSE of model3 (253462.8) is indicative of a better model.

### Now, let's take a look at the impact of a waterfront view on the expected price holding area consta
coef(model3)[2]
```

```
## waterfront
##   821022.9
```

```r
# The expected price would be 861002.3631

### Let's run a multiple regression model on the training set and add some more variables.
#Call this model4. What is the R2 for model4?
model4 = lm(price~bedrooms + bathrooms+ sqft_living + sqft_lot + floors + waterfront + view + condition
pred4 = predict(model4)
sse4 = sum((pred4 - train$price)^2)
sst4 = sum((mean(train$price)-train$price)^2)
model4_r2 = 1 - sse4/sst4
model4_r2
```

```
## [1] 0.6495637
```

```
rmse4 = sqrt(mean((pred4 - train$price)^2))
rmse4
```

```
## [1] 217865.8
```

```
# With a higher R2 (0.6512827) and a lower RMSE (220098.4), model4 is an even better model.

### Let's see which of the predictors used have an influence on price?
summary(model4)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + age, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1333234  -111085    -8889    90391  4197181
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.003e+06  2.060e+04 -48.711  < 2e-16 ***
## bedrooms    -3.797e+04  2.394e+03 -15.857  < 2e-16 ***
## bathrooms    5.337e+04  4.105e+03  13.000  < 2e-16 ***
## sqft_living  1.724e+02  3.887e+00  44.361  < 2e-16 ***
## sqft_lot    -2.442e-01  4.336e-02  -5.632 1.81e-08 ***
## floors       2.415e+04  4.111e+03   5.875 4.32e-09 ***
## waterfront   5.737e+05  2.369e+04  24.211  < 2e-16 ***
## view         4.563e+04  2.705e+03  16.865  < 2e-16 ***
## condition    1.700e+04  2.939e+03   5.785 7.41e-09 ***
## grade        1.220e+05  2.551e+03  47.838  < 2e-16 ***
## age          3.693e+03  8.021e+01  46.035  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 217900 on 15161 degrees of freedom
## Multiple R-squared:  0.6496, Adjusted R-squared:  0.6493
## F-statistic:  2810 on 10 and 15161 DF,  p-value: < 2.2e-16
```

```
# All of the predictors have a significant influence on price.

### Let's day a person decides to add another bathroom. What would be the increase in expected price, h
#   constant?
coef(model4)[3]
```

```
## bathrooms
##  53366.26
```

```
# 50744.76

### Now, out of all the predictors in model4, which exerts the strongest influence on price?
lm.beta(model4)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + age, data = train)
##
## Standardized Coefficients::
## (Intercept)    bedrooms   bathrooms sqft_living    sqft_lot      floors
##  0.00000000 -0.09701320  0.11107546  0.43082385 -0.02763589  0.03540619
##  waterfront        view   condition       grade         age
##  0.12613296  0.09303177  0.03021047  0.38970373  0.29436960
```

```r
# Since sqft_living has the highest beta coefficient, we can say tht it is the strongest predictor of p

### Finally, let's apply this model to test data and calculate what the R2 and RMSE are.
model4 = lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+waterfront+view+condition+grade+age, da
pred4test = predict(model4, newdata = test)
sse4test = sum((pred4test - test$price)^2)
sst4test = sum((mean(test$price)-test$price)^2)
model4test_r2 = 1 - sse4test/sst4test
model4test_r2
```

```
## [1] 0.6588608
```

```r
rmse4test = sqrt(mean((pred4test - test$price)^2))
rmse4test
```

```
## [1] 213162.8
```

```r
# The R2 is slightl higher than in the train sample, with a value of 0.6544801, indicating the model pe
# Similarly, in terms of the RMSE it was slightly lower than the train data, with a value of 207835.2.
# that our model performed well in the test data.
```