Regression HW3 rcd180001

Joshua Durana

2022-09-25

Linear Regression

Linear regression tries to find parameters w and b that minimizes errors in the training data with 2 equations. Algorithms use gradient descent or ordinary least squares to find these parameters. It's a simple algorithm that works best with data with linear patterns. But, it can underfit due to having a high bias towards the shape of the data being linear.

Load Perth housing data and clean data

```
#Load Data
perthCSV <- ("Data/perthHousingData.csv")
perthData <- read.csv(perthCSV, header=TRUE)

#Convert columns into numeric columns
perthData$GARAGE <- as.integer(perthData$GARAGE)
```

```
## Warning: NAs introduced by coercion
```

```
perthData$BUILD_YEAR <- as.integer(perthData$BUILD_YEAR)
```

```
## Warning: NAs introduced by coercion
```

```
#Handle NA values
perthData[perthData == "NULL"] <- NA
perthData$GARAGE[is.na(perthData$GARAGE)] <- 0
perthData$BUILD_YEAR[is.na(perthData$BUILD_YEAR)] <- median(perthData$BUILD_YEAR, na.rm = TRU
E)
perthData$NEAREST_SCH_RANK[is.na(perthData$NEAREST_SCH_RANK)] <- median(perthData$NEAREST_SCH
_RANK, na.rm = TRUE)
```

Divide into train and test sets

Splits the data to train and test to an 80/20 ratio

```
set.seed(920)
perthData <- perthData[3:10]
sampleSize <- floor(.80 * nrow(perthData))
ratio <- sample(seq_len(nrow(perthData)), size = sampleSize)
perthTrain <- perthData[ratio,]
perthTest <- perthData[-ratio,]
```

Data Exploration

```
#Show the first 6 rows of the data frame
head(perthTrain)
```

```
##        PRICE BEDROOMS BATHROOMS GARAGE LAND_AREA FLOOR_AREA BUILD_YEAR CBD_DIST
## 2536  381000        3         2      2       203        126       2014    14700
## 140   532000        4         2      2       653        191       2004    15500
## 9107  390000        3         1      1       716        120       1968    18000
## 33466 565000        4         2      2       626        235       1995    18700
## 7186  515000        4         2      2       703        220       1993    27900
## 14442 450000        3         1      1      2596        114       1954    19300
```

```
#Output the name of all the columns
names(perthTrain)
```

```
## [1] "PRICE"      "BEDROOMS"   "BATHROOMS"  "GARAGE"     "LAND_AREA"
## [6] "FLOOR_AREA" "BUILD_YEAR" "CBD_DIST"
```

```
#Get information on each row
str(perthTrain)
```

```
## 'data.frame':    26924 obs. of  8 variables:
##  $ PRICE     : int  381000 532000 390000 565000 515000 450000 300000 725000 1090000 115000
0 ...
##  $ BEDROOMS  : int  3 4 3 4 4 3 4 4 3 4 ...
##  $ BATHROOMS : int  2 2 1 2 2 1 2 2 2 2 ...
##  $ GARAGE    : num  2 2 1 2 2 1 1 2 1 2 ...
##  $ LAND_AREA : int  203 653 716 626 703 2596 688 2031 334 527 ...
##  $ FLOOR_AREA: int  126 191 120 235 220 114 126 256 135 209 ...
##  $ BUILD_YEAR: int  2014 2004 1968 1995 1993 1954 1968 2004 1930 2001 ...
##  $ CBD_DIST  : int  14700 15500 18000 18700 27900 19300 12300 27900 2700 7400 ...
```

```
#Get the dimensions of the data frame
dim(perthTrain)
```

```
## [1] 26924     8
```

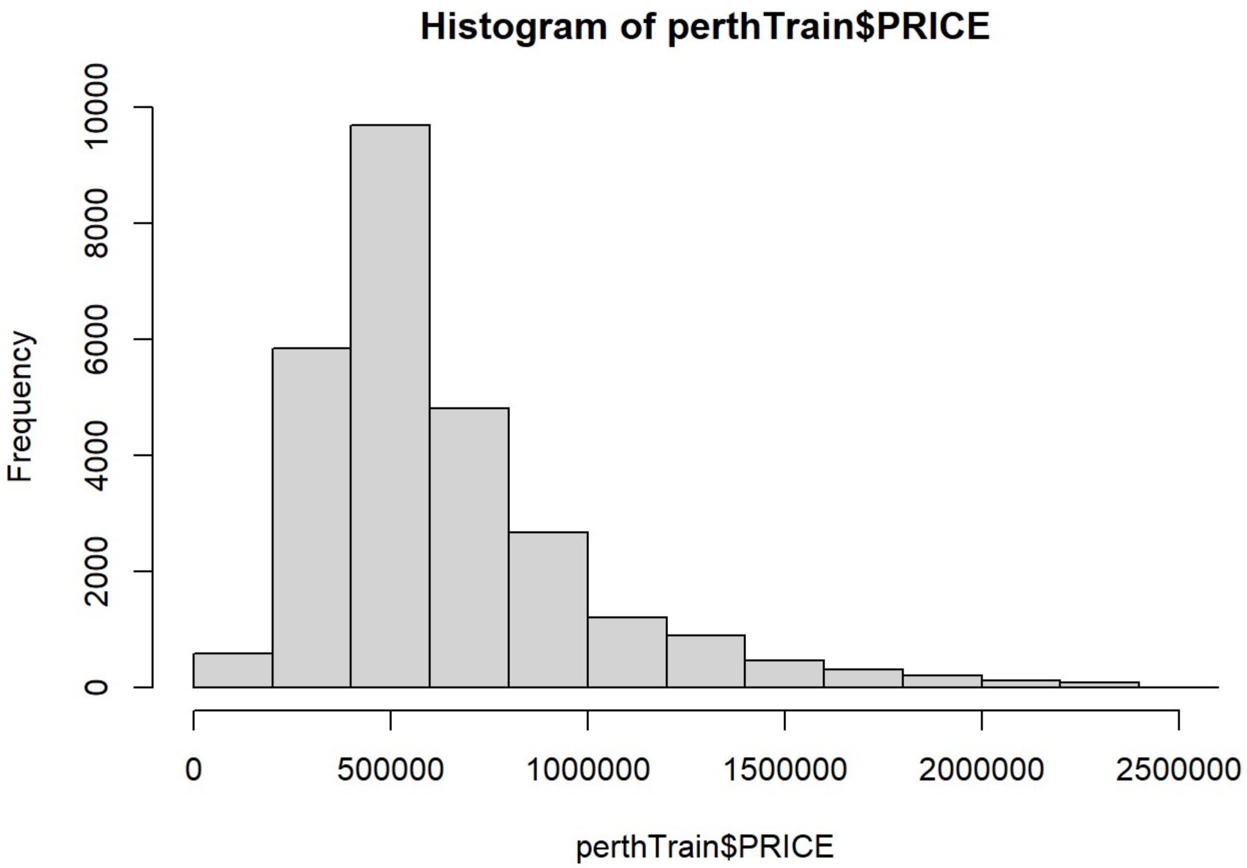
```
#Get the summary of each column
summary(perthTrain)
```

```
##      PRICE            BEDROOMS        BATHROOMS         GARAGE
##  Min.   :  51000   Min.   : 1.000   Min.   :1.000   Min.   : 0.000
##  1st Qu.: 410000   1st Qu.: 3.000   1st Qu.:1.000   1st Qu.: 2.000
##  Median : 535000   Median : 4.000   Median :2.000   Median : 2.000
##  Mean   : 637097   Mean   : 3.661   Mean   :1.824   Mean   : 2.043
##  3rd Qu.: 760000   3rd Qu.: 4.000   3rd Qu.:2.000   3rd Qu.: 2.000
##  Max.   :2440000   Max.   :10.000   Max.   :7.000   Max.   :99.000
##    LAND_AREA         FLOOR_AREA      BUILD_YEAR      CBD_DIST
##  Min.   :    61   Min.   :  1.0   Min.   :1868   Min.   :  681
##  1st Qu.:   503   1st Qu.:130.0   1st Qu.:1980   1st Qu.:11100
##  Median :   682   Median :172.0   Median :1995   Median :17500
##  Mean   :  2660   Mean   :183.4   Mean   :1990   Mean   :19752
##  3rd Qu.:   836   3rd Qu.:222.0   3rd Qu.:2004   3rd Qu.:26600
##  Max.   :999999   Max.   :870.0   Max.   :2017   Max.   :59800
```

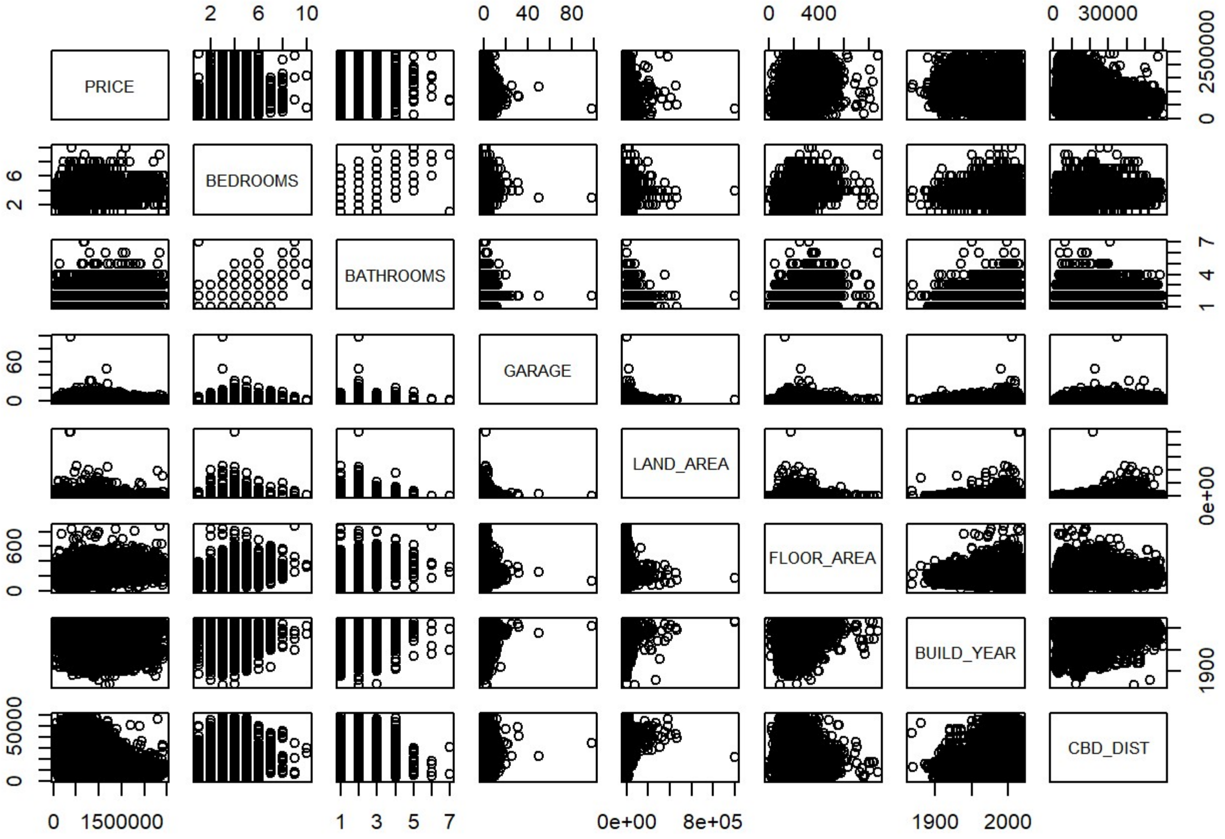
```
#Get the correlations between Price, Bedrooms, Garage, Bathrooms, Land Area, Floor Area, and
the Distance to the Business District
cor(perthTrain)
```

```
##                  PRICE   BEDROOMS  BATHROOMS     GARAGE    LAND_AREA FLOOR_AREA
## PRICE       1.0000000 0.25354226 0.38307982 0.13497875  0.059636998 0.55305560
## BEDROOMS    0.2535423 1.00000000 0.56200796 0.19356306  0.051044708 0.53896064
## BATHROOMS   0.3830798 0.56200796 1.00000000 0.20878454  0.029052168 0.56118582
## GARAGE      0.1349788 0.19356306 0.20878454 1.00000000  0.041445085 0.17135518
## LAND_AREA   0.0596370 0.05104471 0.02905217 0.04144509  1.000000000 0.07896021
## FLOOR_AREA  0.5530556 0.53896064 0.56118582 0.17135518  0.078960210 1.00000000
## BUILD_YEAR -0.1516801 0.21724470 0.32848858 0.05078001 -0.000171502 0.21502247
## CBD_DIST   -0.3578310 0.12175558 0.03316057 0.03056826  0.142857089 0.01807537
##              BUILD_YEAR    CBD_DIST
## PRICE      -0.151680148 -0.35783104
## BEDROOMS    0.217244702  0.12175558
## BATHROOMS   0.328488580  0.03316057
## GARAGE      0.050780014  0.03056826
## LAND_AREA  -0.000171502  0.14285709
## FLOOR_AREA  0.215022467  0.01807537
## BUILD_YEAR  1.000000000  0.25644153
## CBD_DIST    0.256441529  1.00000000
```

```
#Histogram of the price
hist(perthTrain$PRICE)
```



```
#Plots of multiple columns
pairs(perthTrain)
```



Linear Regression

I will use bedroom as my predictor for the price

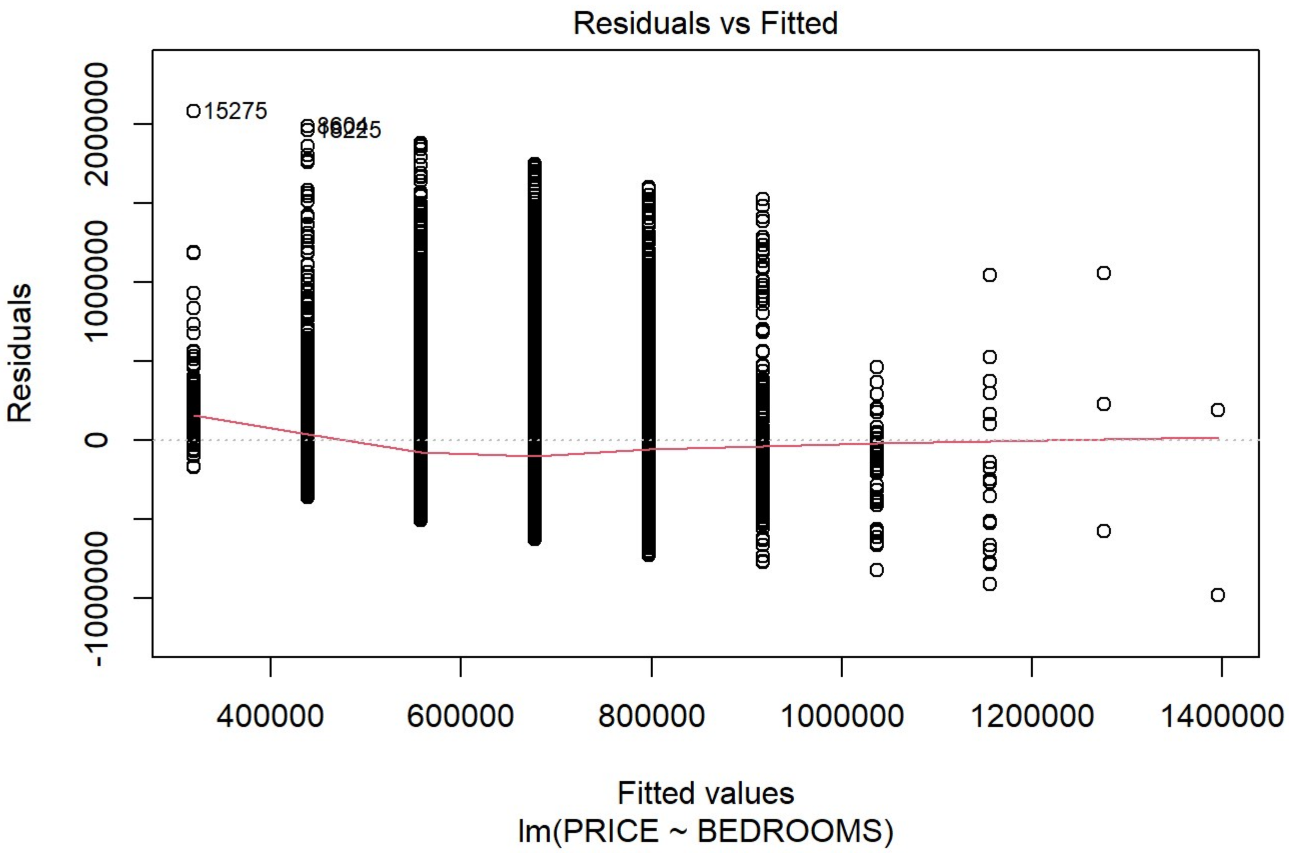
```
simplePerth <- lm(PRICE~BEDROOMS, data=perthTrain)
summary(simplePerth)
```

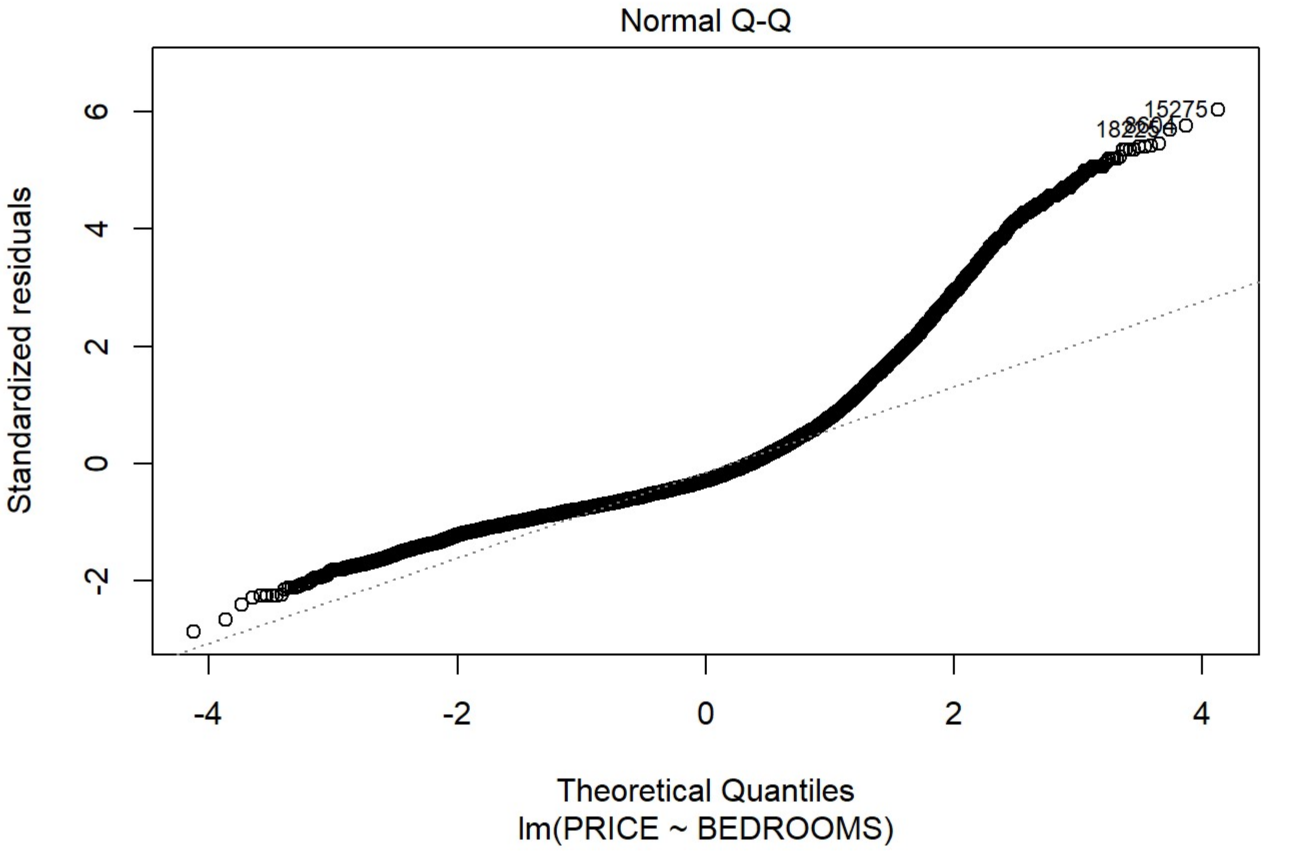
```
## 
## Call:
## lm(formula = PRICE ~ BEDROOMS, data = perthTrain)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -986532 -218025  -98025  120759 2081263 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   199093      10399   19.14   <2e-16 ***
## BEDROOMS      119644       2782   43.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 345000 on 26922 degrees of freedom
## Multiple R-squared:  0.06428,    Adjusted R-squared:  0.06425 
## F-statistic:  1850 on 1 and 26922 DF,  p-value: < 2.2e-16
```

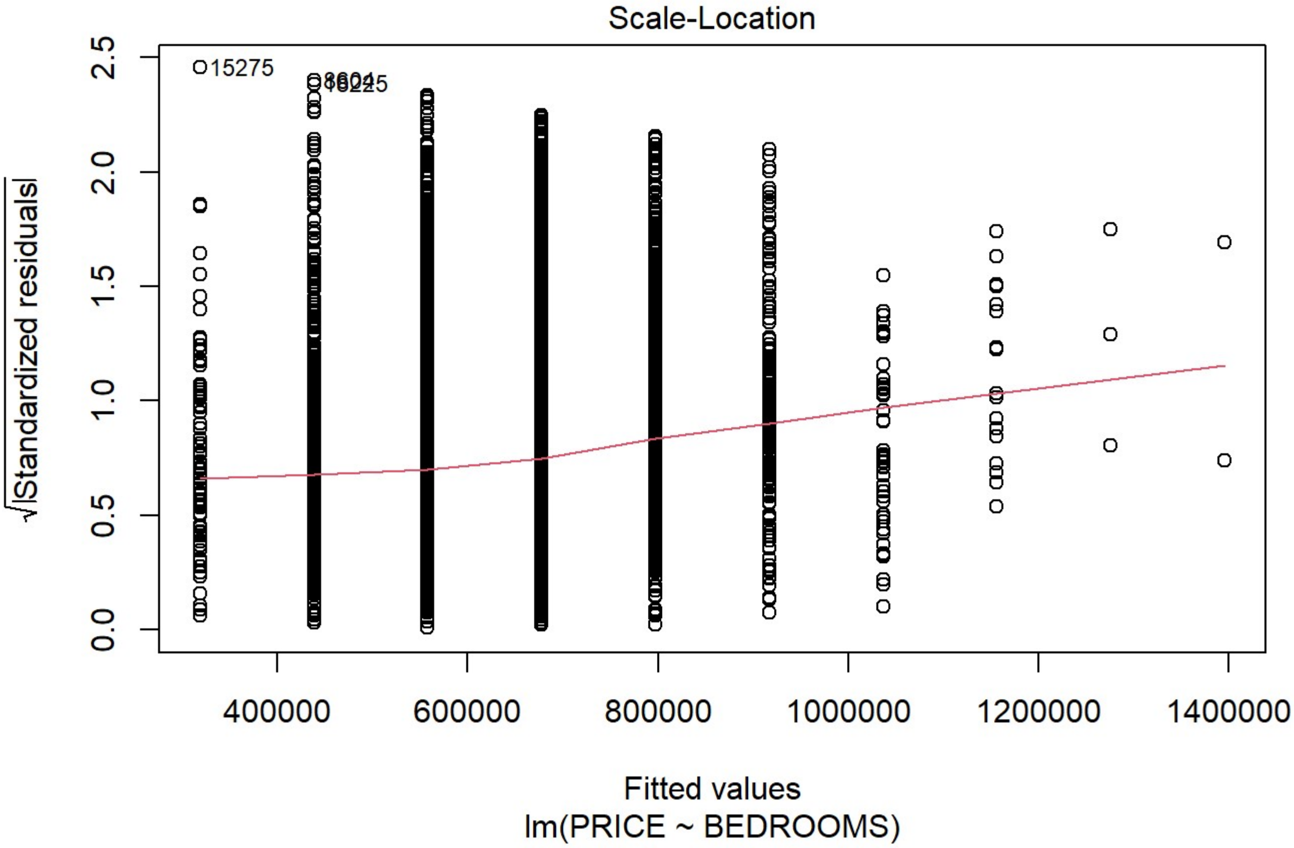
The estimate and intercept gives us the actual linear formula to predict price. As bedrooms increase the price will increase by $119644. The intercept if only for fitting the data and is just added with the quotient of the number of bedrooms and the estimate.The residual standard error (RSE) tells us how off our model is with the training data. Our model predicts the price of a house with an average error of $345000. The R-squared value is valued from 0 to 1 and measures how variance is explained by the predictors as it gets closer to 1. The value is .064 which is not very good. The F statistic tells us whether the amount of bedrooms is a significant predictor of price, we want a F-statistic greater than 1 and a low p-value. The F-statistic is greater than 1 and the p-value is low, so that means the amount of bedrooms is a significant predictor of price.

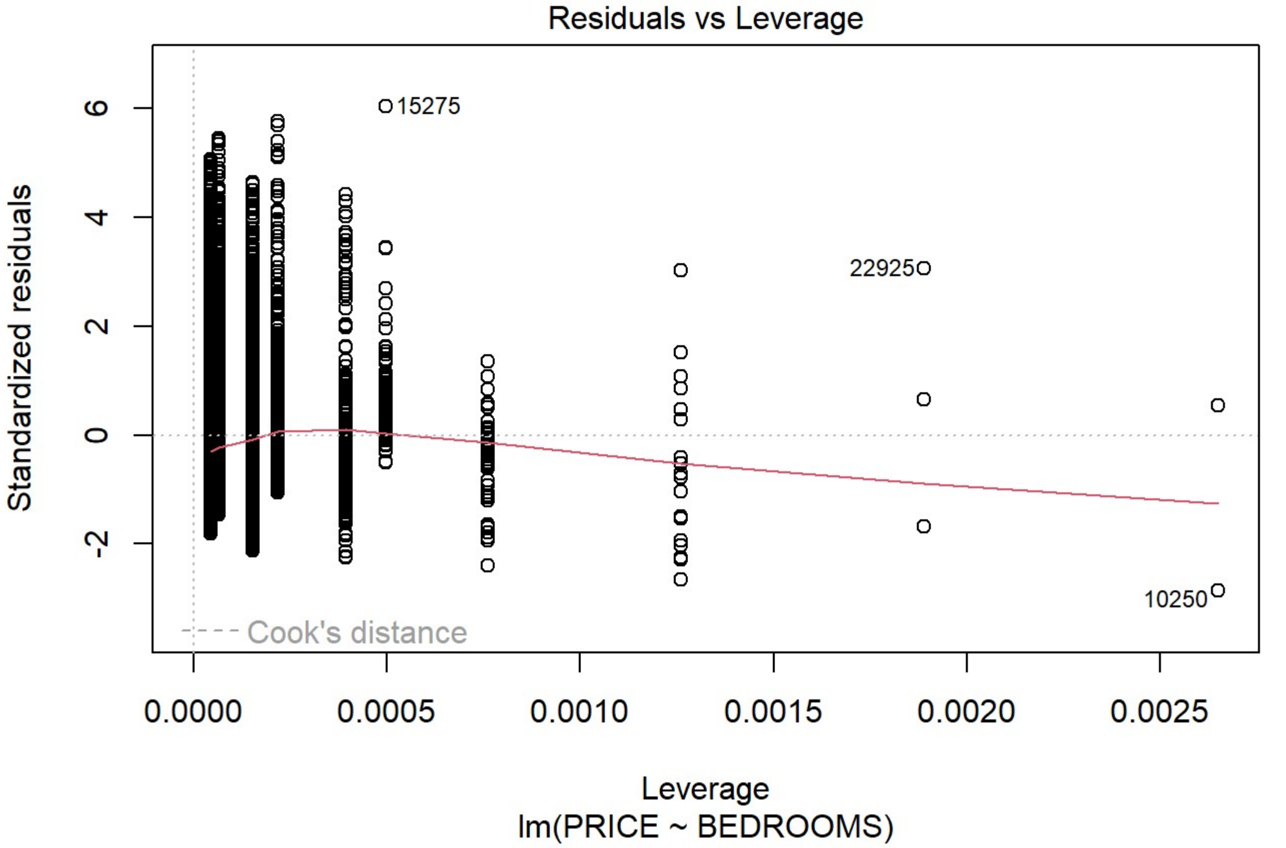
Residual Plots

```
plot(simplePerth)
```


Residuals vs Fitted
15275
8604
18225
Residuals
-1000000
0
1000000
2000000
400000
600000
800000
1000000
1200000
1400000
Fitted values
lm(PRICE ~ BEDROOMS)


Normal Q-Q
15275
18225
8604
Standardized residuals
-2
0
2
4
6
-4
-2
0
2
4
Theoretical Quantiles
lm(PRICE ~ BEDROOMS)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
15275
8604
16225
400000
600000
800000
1000000
1200000
1400000
Fitted values
lm(PRICE ~ BEDROOMS)


Residuals vs Leverage
Standardized residuals
-2
0
2
4
6
15275
22925
10250
Cook's distance
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
Leverage
lm(PRICE ~ BEDROOMS)

The Residuals vs Fitted plot shows whether the residuals have a non-linear pattern. A horizontal line with residuals equally spread over it shows the model doesn’t have any non-linear relationships. The line is pretty horizontal, but the spread of the residuals is concentrated on the left side. This shows that there’s a non-linear relationship in the model.

The Normal Q-Q plot shows if the residuals are normally distribute, which is shown as a straight diagonal line. The line deviates at the right side, which shows that the residuals are not normally distributed.

The Scale-Location plot shows if the residuals are spread equally along the ranges of predictors, which is shown as a straight line with the residuals spread equally around it. The line moves up towards the right, and the residuals are spread towards the left. This shows that the residuals aren’t spread equally along the predictors.

The Residuals vs Leverage plot shows if there are any influential outliers in the data. This is shown whether there are outlying values in the top or bottom right hand corner and cases outside of th Cook’s distance which is shown with a dashed line. There are not cases outside of the Cook’s distance. This shows that there are not influential outliers.

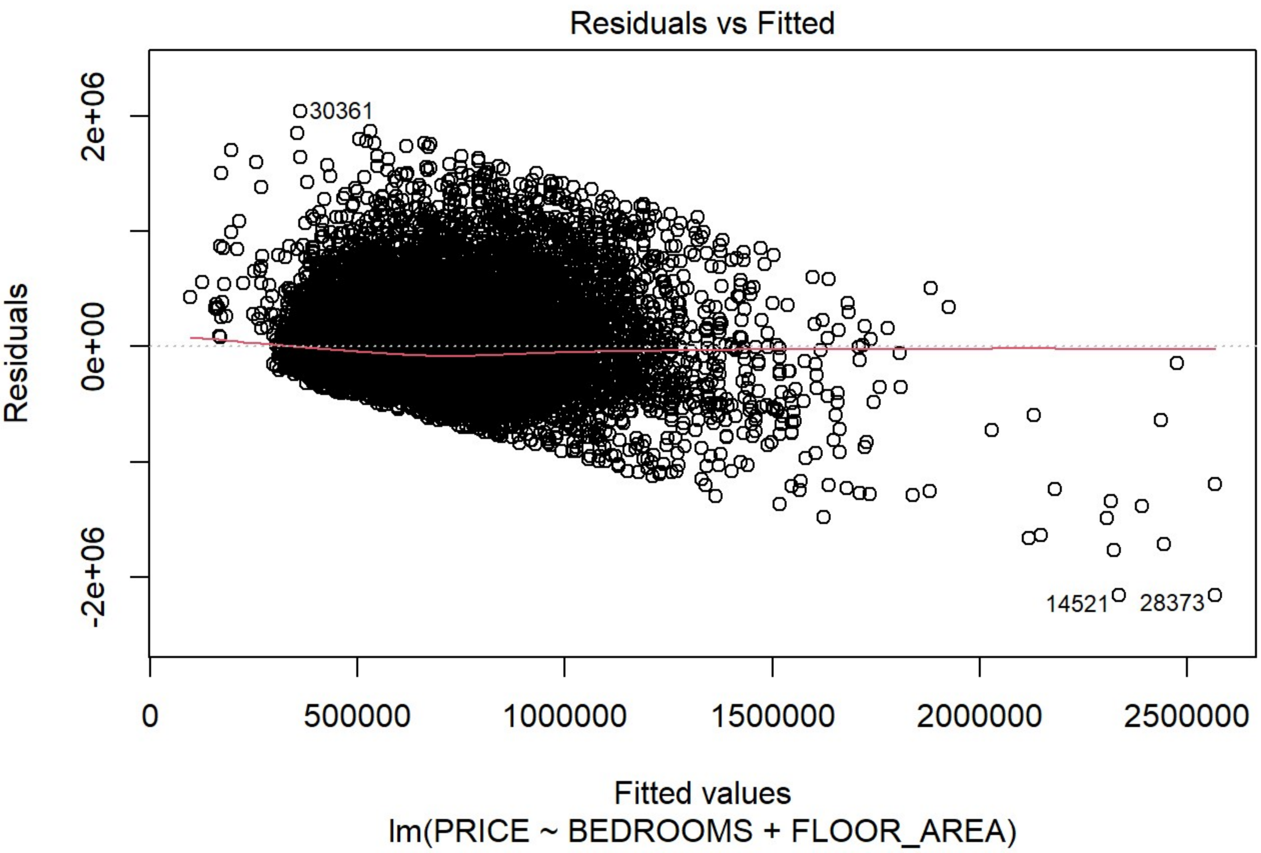
Multiple Linear Regression

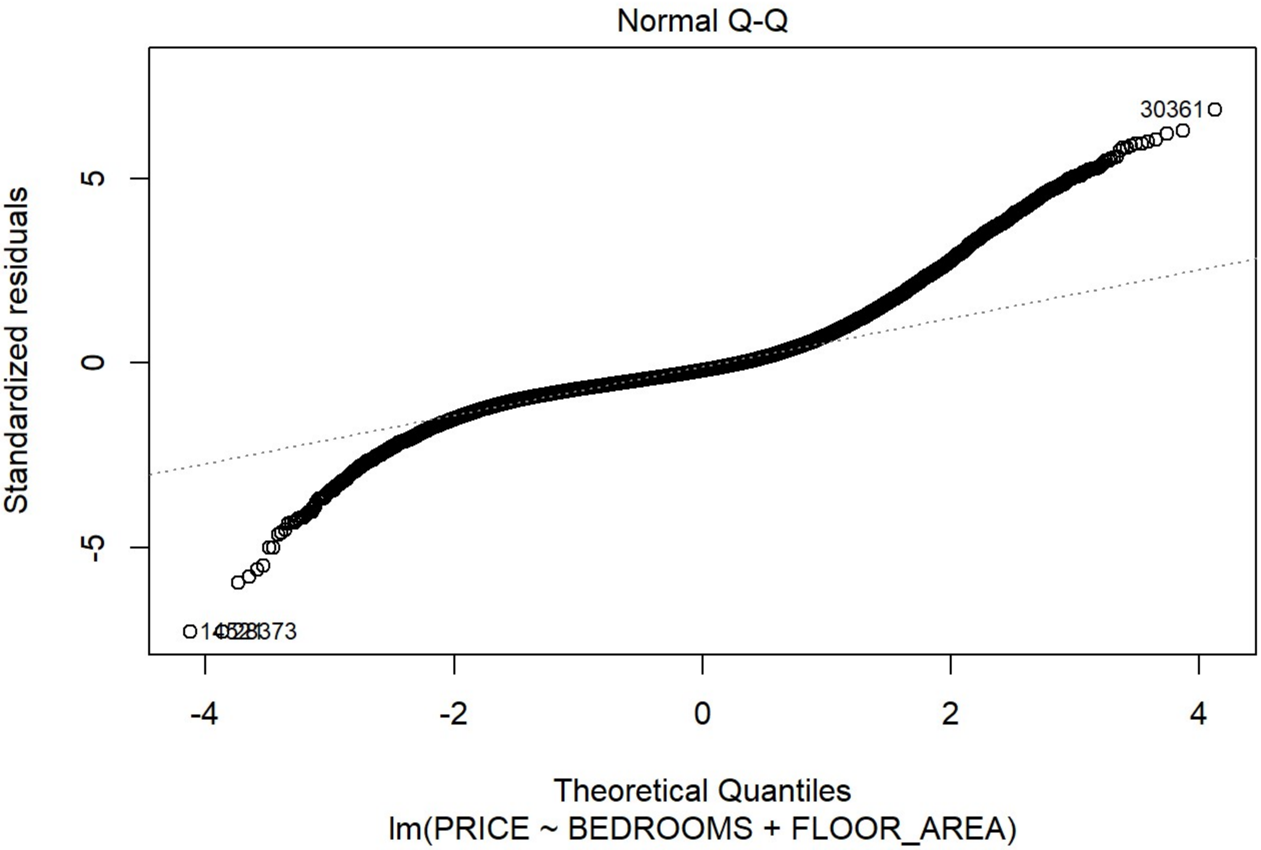
I will use both the number of bedrooms and the floor area to predict price

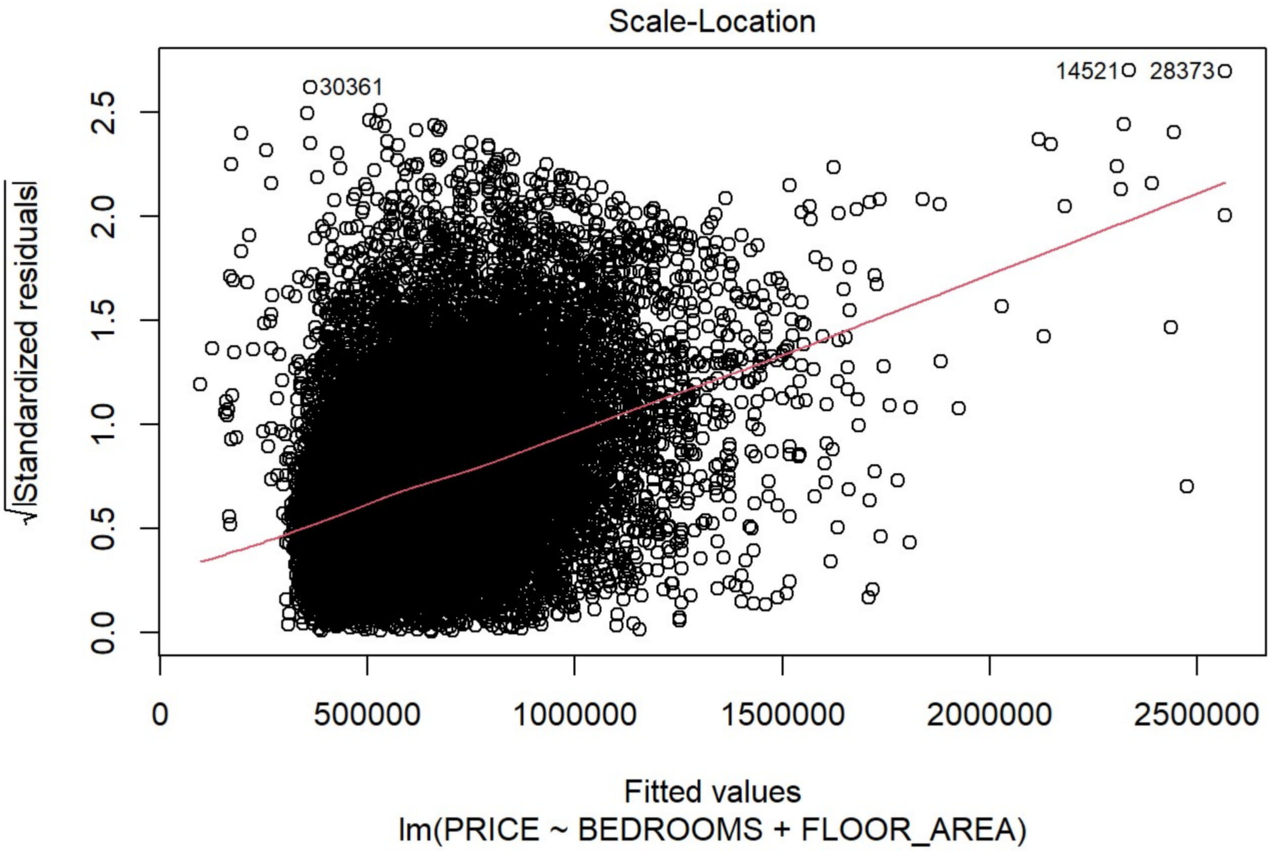
```
multPerth <- lm(PRICE~BEDROOMS+FLOOR_AREA, data = perthTrain)
summary(multPerth)
```

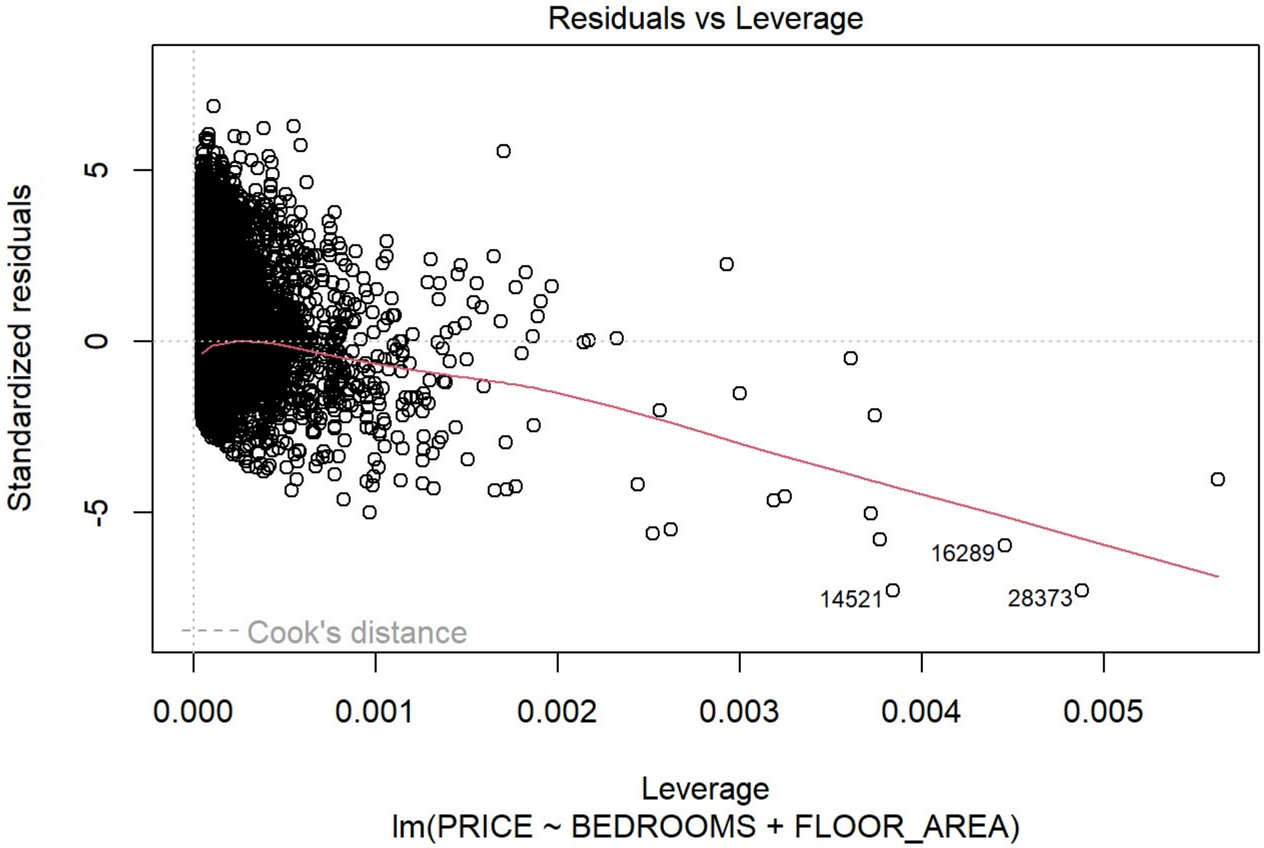
```
##
## Call:
## lm(formula = PRICE ~ BEDROOMS + FLOOR_AREA, data = perthTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2159415  -163779   -61369    98495  2038134
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 212304.0     8940.0   23.75   <2e-16 ***
## BEDROOMS    -29618.0     2838.9  -10.43   <2e-16 ***
## FLOOR_AREA    2907.5       29.8   97.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 296600 on 26921 degrees of freedom
## Multiple R-squared:  0.3087, Adjusted R-squared:  0.3086
## F-statistic:  6010 on 2 and 26921 DF,  p-value: < 2.2e-16
```

```
plot(multPerth)
```


Residuals vs Fitted
Residuals
2e+06
0e+00
-2e+06
30361
14521
28373
0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(PRICE ~ BEDROOMS + FLOOR_AREA)


Normal Q-Q
Standardized residuals
5
0
-5
30361
14521
28373
-4
-2
0
2
4
Theoretical Quantiles
lm(PRICE ~ BEDROOMS + FLOOR_AREA)


Scale-Location
30361
14521
28373
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(PRICE ~ BEDROOMS + FLOOR_AREA)

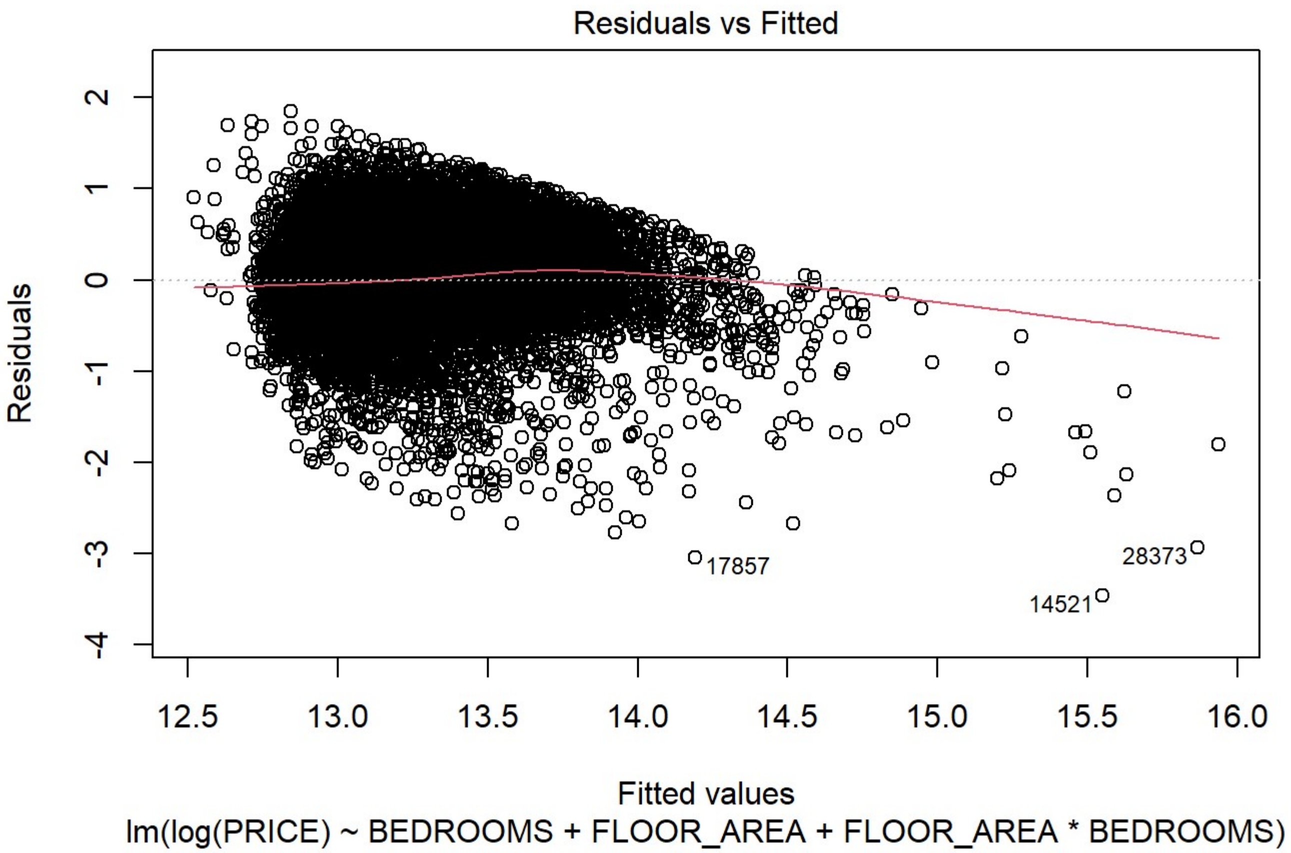

Residuals vs Leverage
Standardized residuals
5
0
-5
16289
14521
28373
Cook's distance
0.000
0.001
0.002
0.003
0.004
0.005
Leverage
lm(PRICE ~ BEDROOMS + FLOOR_AREA)

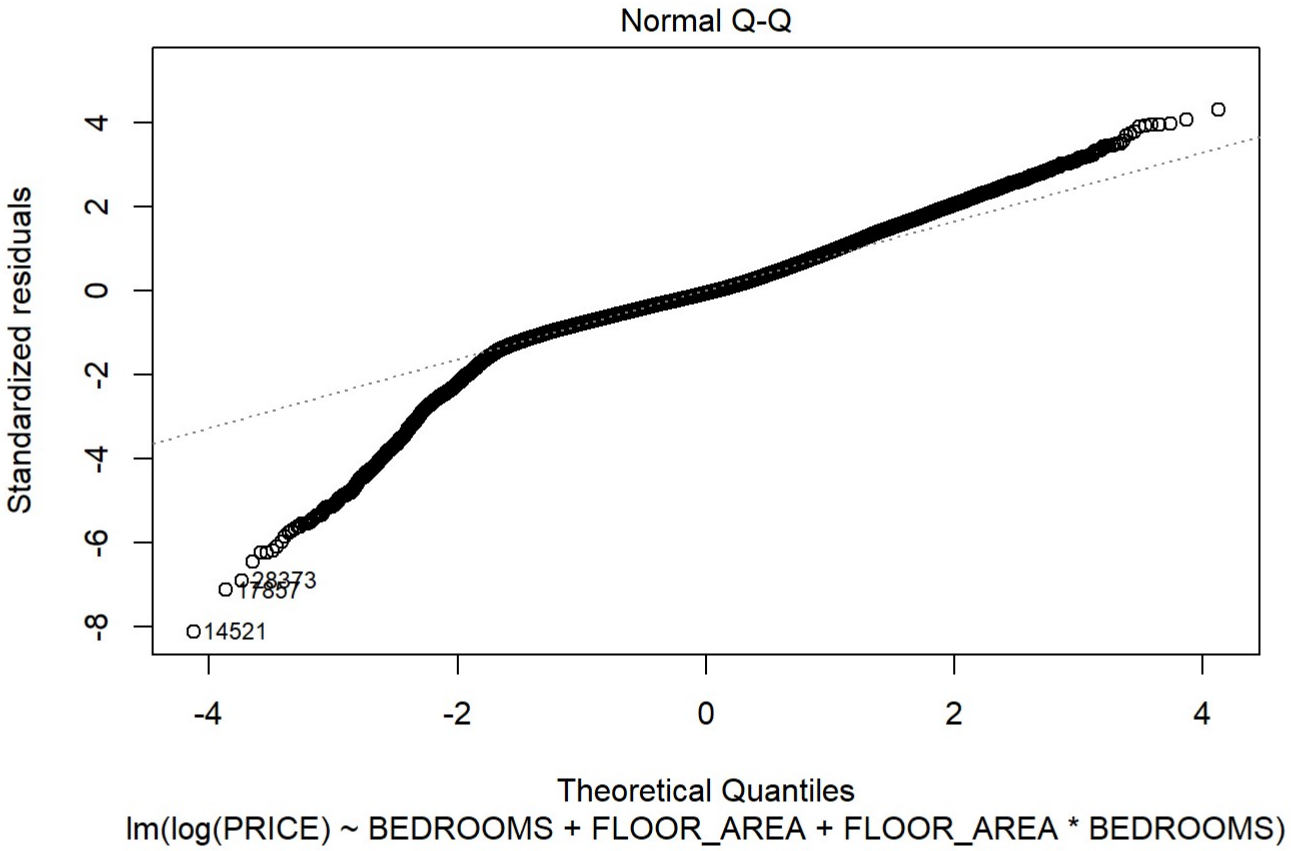
Third Linear Regression

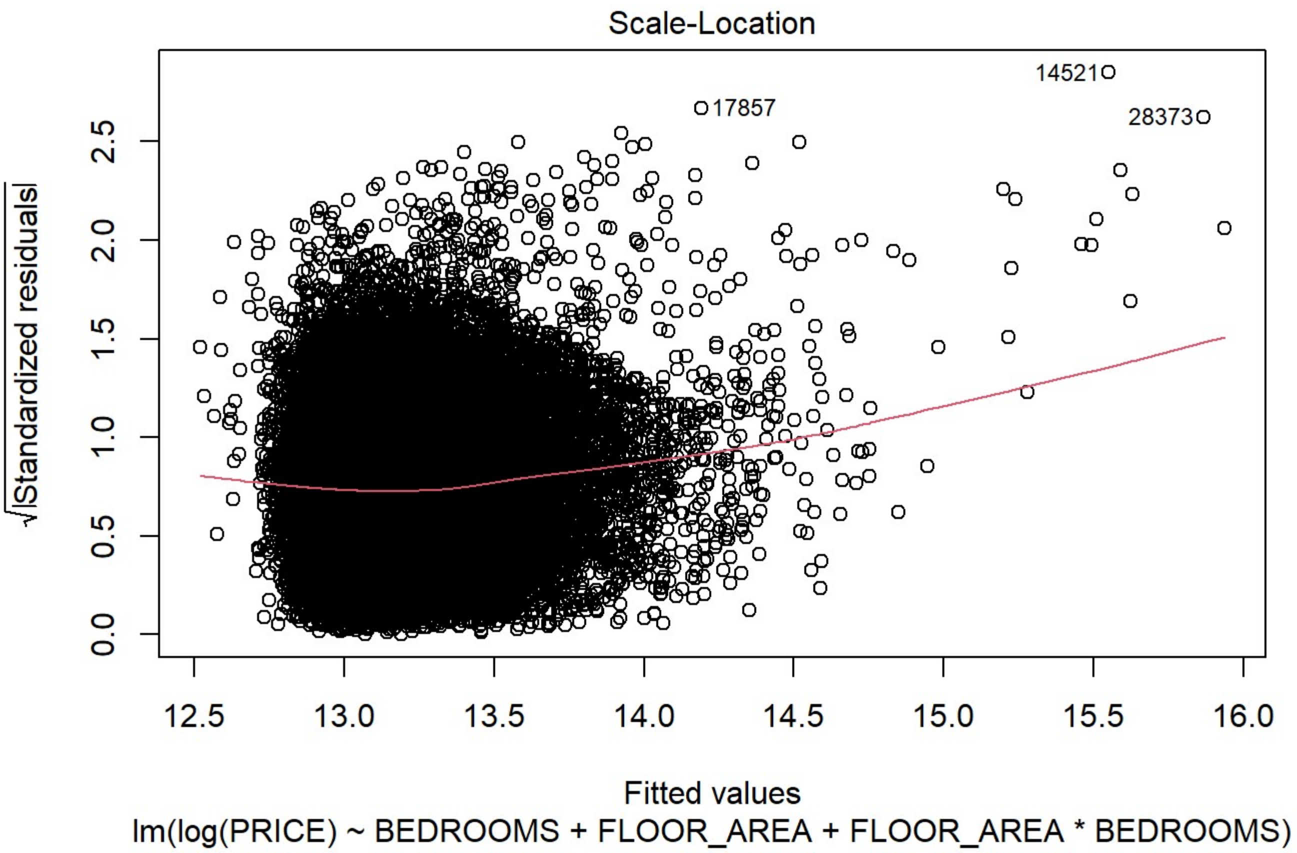
```
perthModel <- lm(log(PRICE)~BEDROOMS+FLOOR_AREA+FLOOR_AREA*BEDROOMS, data=perthTrain)
summary(perthModel)
```

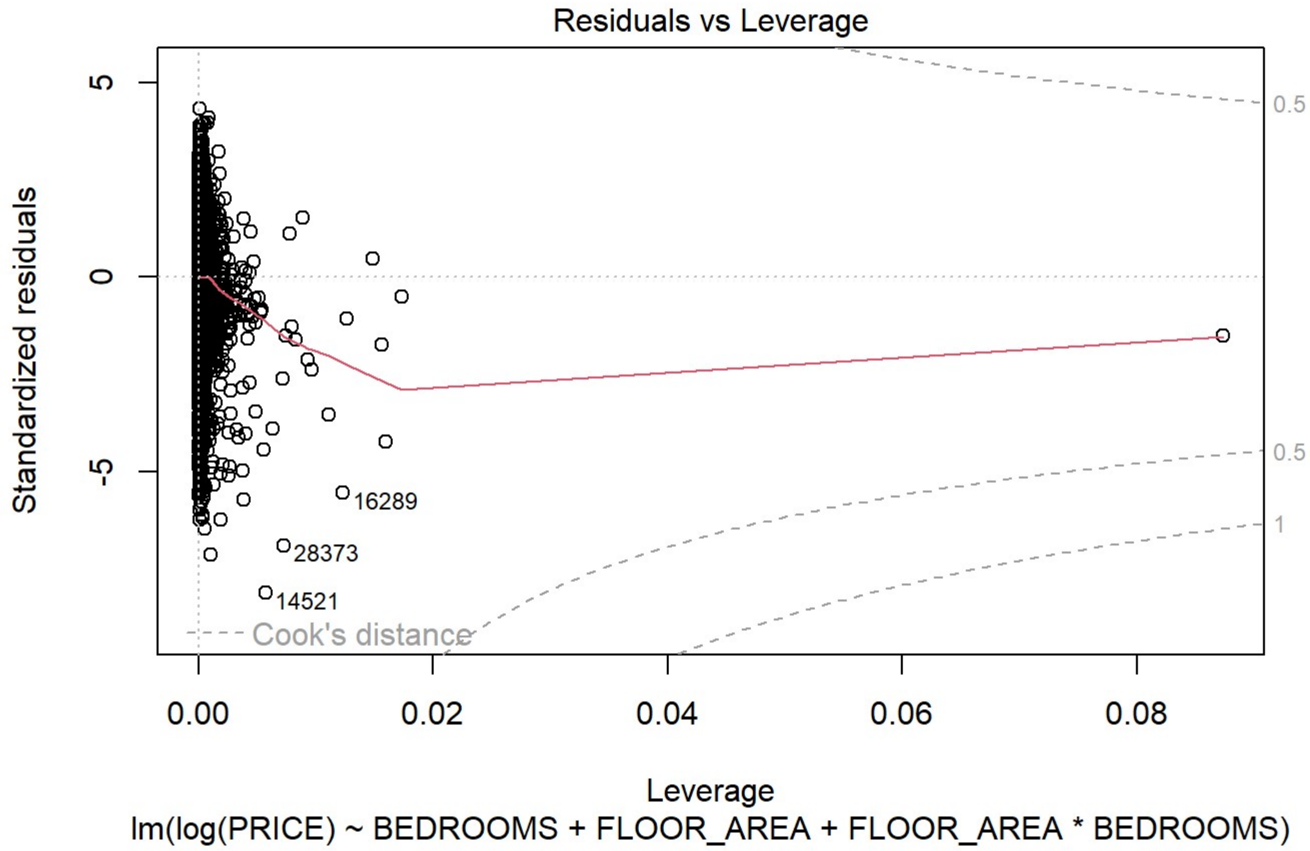
```
##
## Call:
## lm(formula = log(PRICE) ~ BEDROOMS + FLOOR_AREA + FLOOR_AREA *
##     BEDROOMS, data = perthTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4690 -0.2305 -0.0264  0.2429  1.8491
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         12.4688805  0.0278399 447.878  < 2e-16 ***
## BEDROOMS             0.0152841  0.0078354   1.951   0.0511 .
## FLOOR_AREA           0.0044474  0.0001438  30.929  < 2e-16 ***
## BEDROOMS:FLOOR_AREA -0.0001525  0.0000356  -4.285 1.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4282 on 26920 degrees of freedom
## Multiple R-squared:  0.2887, Adjusted R-squared:  0.2886
## F-statistic:  3642 on 3 and 26920 DF,  p-value: < 2.2e-16
```

```
plot(perthModel)
```


Residuals vs Fitted
Residuals
2
1
0
-1
-2
-3
-4
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
17857
28373
14521
Fitted values
lm(log(PRICE) ~ BEDROOMS + FLOOR_AREA + FLOOR_AREA * BEDROOMS)


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-6
-8
-4
-2
0
2
4
28373
17857
14521
Theoretical Quantiles
lm(log(PRICE) ~ BEDROOMS + FLOOR_AREA + FLOOR_AREA * BEDROOMS)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
14521
17857
28373
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
Fitted values
lm(log(PRICE) ~ BEDROOMS + FLOOR_AREA + FLOOR_AREA * BEDROOMS)


Residuals vs Leverage
Standardized residuals
5
0
-5
0.5
0.5
1
16289
28373
14521
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
lm(log(PRICE) ~ BEDROOMS + FLOOR_AREA + FLOOR_AREA * BEDROOMS)

Comparing Models

All three models have good predictors shown by the 3 stars on each of the predictors for each model. Comparing the R-squared metric, the second model’s R-squared metric was better than the other two.

In the Residuals vs Fitted plots all three were very similar, their lines were pretty horizontal and the residuals are mainly on the right side of the graph. This shows that there’s a non-linear relationship between the predictors and outcome variables.

The Normal Q-Q plots for the first and second model were pretty similar, they both curved on the right side of the graph. For the third model, it mainly curved on the left side then it straightened out on the right side. This shows that the training data has a high amount of extreme values.

The Scale-Location plots for the first and third model are pretty similar where the points are clustered on the right side and the line slightly curves up. The second model’s points are similar to the first and third, but the line has a steep angle. This shows that our data isn’t homoscedastic or have equal variance.

The Residuals vs Leverage plot for the first two models are very similar, all their cases are within the Cook’s distance lines. The third model has one case that’s much further out on the leverage, but is within the Cook’s distance lines. This shows that there aren’t any influential outliers on the dataset.

Testing

```
#Model 1
pred1 <- predict(simplePerth, newdata=perthTest)
cor(pred1, perthTest$PRICE)
```

```
## [1] 0.2489104
```

```
mean((pred1-perthTest$PRICE)^2)
```

```
## [1] 116533349341
```

```
sqrt(mean((pred1-perthTest$PRICE)^2))
```

```
## [1] 341369.8
```

```
#Model 2
pred2 <- predict(multPerth, newdata=perthTest)
cor(pred2, perthTest$PRICE)
```

```
## [1] 0.5314572
```

```
mean((pred2-perthTest$PRICE)^2)
```

```
## [1] 89311492527
```

```
sqrt(mean((pred2-perthTest$PRICE)^2))
```

```
## [1] 298850.3
```

```
#Model 3
pred3 <- predict(perthModel, newdata=perthTest)
cor(pred3, perthTest$PRICE)
```

```
## [1] 0.5293419
```

```
mean((pred3-perthTest$PRICE)^2)
```

```
## [1] 529944754006
```

```
sqrt(mean((pred3-perthTest$PRICE)^2))
```

```
## [1] 727973
```

Comparing the correlation metric between all 3 models, the second and third model were much better than the first model. Comparing the second and third models for correlation, the second model the best by a small margin because it’s closest to 1.

Comparing the mean squared regression (MSE), the third model was the best out of all the models since it has the lowest MSE value.