Multiple Linear Regression Case Study: Boston Housing Prices

Background: The data has 506 cases where each case is a location in Boston. The “median housing price” is a target variable. The data has many other variables related to environment,education,,crime etc.which can influence the housing prices in the specific location

The objective is to identify significant factors affecting housing prices.

Import data and display first 6 rows

```
hp<-read.csv("Housing Prices.csv",header=T)
head(hp)
```

```
     CRIM ZN INDUS CHAS   NOX    RM  AGE    DIS RAD TAX PTRATIO LSTAT MEDV
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

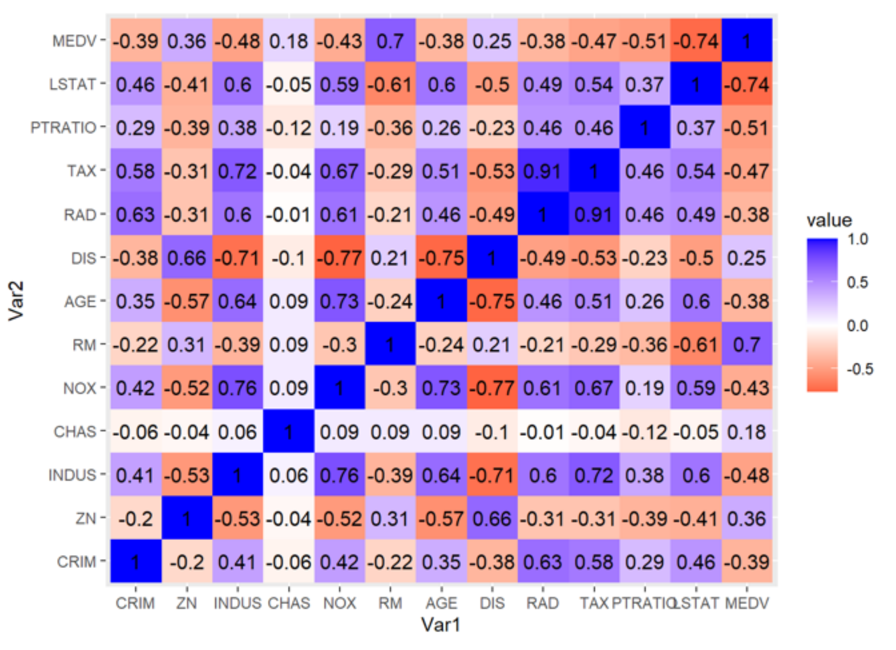
Data Description

Column name	Column description
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 in sq ft
INDUS	proportion of non-retail business acres per town %
CHAS	Charles River dummy variable (=1 if tract bounds river;0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940 %
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

Correlation matrix using heatmap

```
cormat <- round(cor(hp),2)
melted_cormat <- melt(cormat)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()+
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 4)+
  scale_fill_gradient2(low="red",mid="white",high="blue")
```



Develop Housing Prices Model using Linear Regression

Split original data into training and testing data sets

```
library(caret)
index<-createDataPartition(hp$MEDV,p=0.8,list=FALSE)

traindata<-hp[index,]
testdata<-hp[-index,]
```

Linear Regression using lm function

```
hp_model<-lm(MEDV~CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+TAX+PTRATIO+LSTAT,data=traindata)
hp_model
```

```

Call:
lm(formula = MEDV ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + 
    DIS + RAD + TAX + PTRATIO + LSTAT, data = traindata)

Coefficients:
(Intercept)         CRIM           ZN        INDUS         CHAS          NOX  
  40.583521    -0.124911     0.057102     0.006881     3.500089   -16.064356  
         RM          AGE          DIS          RAD          TAX      PTRATIO  
   3.497604    -0.008060    -1.418626     0.328398    -0.013573    -0.902038  
      LSTAT  
  -0.526081  
```

Display parameter estimates with other model statistics

```
summary(hp_model)
```

```

Call:
lm(formula = MEDV ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + 
    DIS + RAD + TAX + PTRATIO + LSTAT, data = traindata)

Residuals:
   Min     1Q Median     3Q    Max 
-9.975 -2.857 -0.468  1.722 26.448 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  40.583521   5.405317   7.508 4.05e-13 ***
CRIM         -0.124911   0.034976  -3.571 0.000399 ***
ZN            0.057102   0.014922   3.827 0.000151 ***
INDUS         0.006881   0.069719   0.099 0.921434    
CHAS          3.500089   0.983235   3.560 0.000417 ***
NOX         -16.064356   4.226263  -3.801 0.000167 ***
RM            3.497604   0.460647   7.593 2.29e-13 ***
AGE          -0.008060   0.015049  -0.536 0.592516    
DIS          -1.418626   0.219882  -6.452 3.25e-10 ***
RAD           0.328398   0.073917   4.443 1.16e-05 ***
TAX          -0.013573   0.004214  -3.221 0.001383 ** 
PTRATIO      -0.902038   0.143135  -6.302 7.87e-10 ***
LSTAT        -0.526081   0.055287  -9.516  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.762 on 394 degrees of freedom
Multiple R-squared:  0.7378,	Adjusted R-squared:  0.7298 
F-statistic: 92.38 on 12 and 394 DF,  p-value: < 2.2e-16
```

Comment: “INDUS” and “AGE” are only insignificant variables.

The model explains 74% of variation in dependent variable “MEDV”(R^2=0.7378)

Check for multicollinearity using vif function

```
library(car)
vif(hp_model)
```

```
    CRIM       ZN    INDUS     CHAS      NOX       RM      AGE      DIS 
1.807258 2.335490 4.148096 1.074867 4.421034 1.920367 3.158816 4.000806 
     RAD      TAX  PTRATIO    LSTAT 
7.238041 8.857655 1.790769 2.891889 
```

Comment: It is observed that variable TAX has high vif

The variable “TAX” is excluded

```
hp_model1<-lm(MEDV~CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+PTRATIO+LSTAT,data=traindata)
```

Check for multicollinearity again

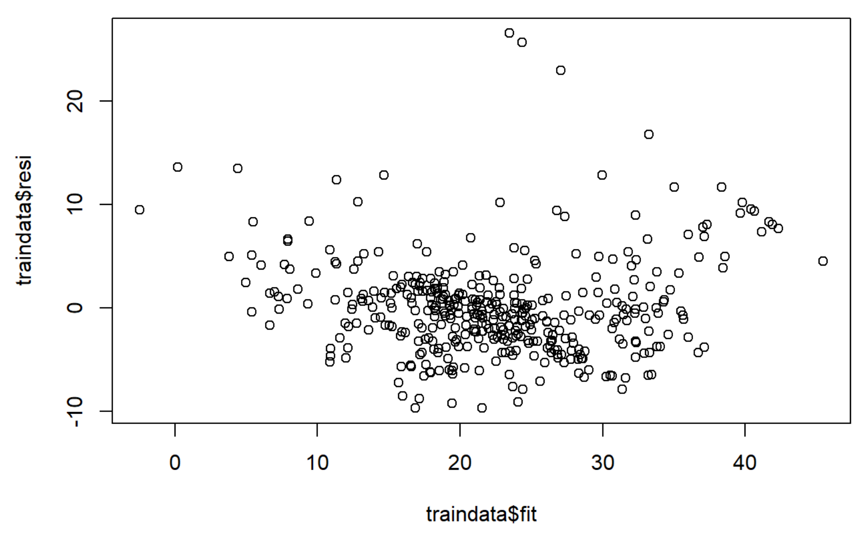
```
library(car)
vif(hp_model1)
```

```
    CRIM       ZN    INDUS     CHAS      NOX       RM      AGE      DIS 
1.807173 2.210505 3.327253 1.053989 4.404969 1.911054 3.146029 3.996038 
     RAD  PTRATIO    LSTAT 
2.749241 1.783768 2.891849 
```

Comment: The multicollinearity problem is resolved as all VIF’s are less than 5

Plot of Residuals vs Predicted values

```
traindata$fit<-fitted(hp_model1)
traindata$resi<-residuals(hp_model1)
plot(traindata$fit,traindata$resi)
```

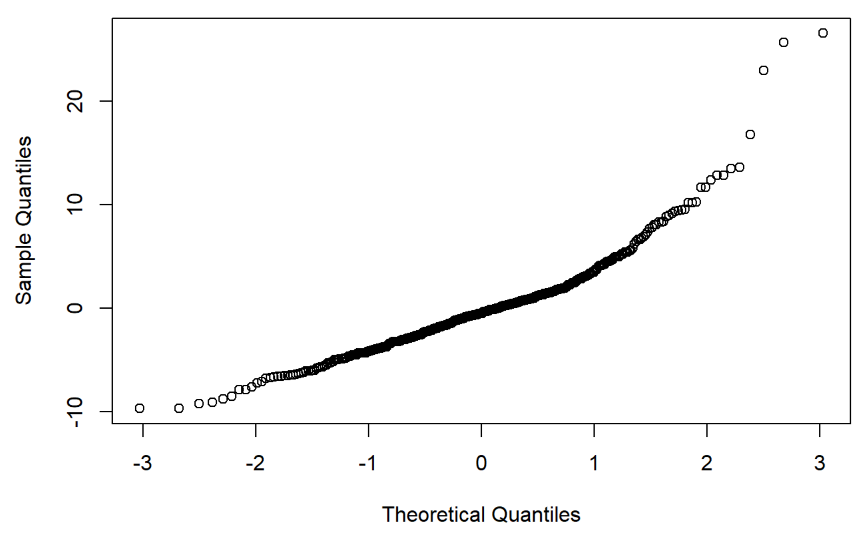


Comment: It is observed that residuals are randomly distributed and uncorelated with predicted values

Check if distribution of errors is “NORMAL”

```
qqnorm(traindata$resi)
```

Normal Q-Q Plot




```
shapiro.test(traindata$resi)
```

```

	Shapiro-Wilk normality test

data:  traindata$resi
W = 0.90658, p-value = 3.932e-15
```

```
library(nortest)
lillie.test(traindata$resi)
```

```

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  traindata$resi
D = 0.11208, p-value = 2.386e-13
```

Comment: Although normality of errors is not established we will proceed to evaluate the model performance

Rerun the model after removing the insignificant variables

```
hp_model2<-lm(MEDV~CRIM+ZN+CHAS+NOX+RM+DIS+RAD+PTRATIO+LSTAT,data=traindata)
summary(hp_model2)
```

```

Call:
lm(formula = MEDV ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + 
    PTRATIO + LSTAT, data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.6931 -3.0972 -0.3564  1.6947 26.1051 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  39.15489    5.43287   7.207 2.90e-12 ***
CRIM         -0.12159    0.03539  -3.436 0.000653 ***
ZN            0.04750    0.01459   3.255 0.001231 ** 
CHAS          3.76958    0.98017   3.846 0.000140 ***
NOX         -20.06377    3.81373  -5.261 2.35e-07 ***
RM            3.64436    0.45351   8.036 1.07e-14 ***
DIS          -1.32353    0.20647  -6.410 4.12e-10 ***
RAD           0.13594    0.04561   2.981 0.003054 ** 
PTRATIO      -0.98204    0.14138  -6.946 1.55e-11 ***
LSTAT        -0.54551    0.05311 -10.271  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.823 on 397 degrees of freedom
Multiple R-squared:  0.729,	Adjusted R-squared:  0.7228 
F-statistic: 118.7 on 9 and 397 DF,  p-value: < 2.2e-16
```

Calculate RMSE values based on residuals using first principle

```
traindata$resi<-residuals(hp_model2)

RMSE<-sqrt(mean(traindata$resi**2))
RMSE
```

```
[1] 4.762943
```

Model Validation: Holdout Method using RMSE

```
testdata$pred<-predict(hp_model2,testdata)
testdata$res<-(testdata$MEDV-testdata$pred)
RMSEtest<-sqrt(mean(testdata$res**2))
RMSEtest
```

```
[1] 5.040017
```

K-fold cross validation using caret package

```
library(caret)
kfolds<-trainControl(method="cv",number=4)
kmodel<- train(MEDV~CRIM+ZN+CHAS+NOX+RM+DIS+RAD+PTRATIO+LSTAT,data=hp,method="lm",
              trControl=kfolds)
kmodel
```

```
Linear Regression 

506 samples
  9 predictor

No pre-processing
Resampling: Cross-Validated (4 fold) 
Summary of sample sizes: 380, 379, 380, 379 
Resampling results:

  RMSE      Rsquared   MAE    
  4.958156  0.7157767  3.49117

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Comment: RMSE and R squared values using K-fold validation are similar to overall RMSE and R squared values

The model can be implemented for decision making