# Boston Housing Regression Analysis

Joseph Bruno

2025-04-04

## Introduction

### Objective

The overall objective of this analysis is to identify factors influencing housing prices in Boston.

### Dataset

The Boston Housing dataset is a popular dataset used for statistical modeling, included in the MASS R library. It contains information on 506 houses, with the following 14 variables:
- **crim** - Crime rate by town
- **zn** - Proportion of residential land zoned for large lots
- **indus** - Proportion of non-retail business acres per town
- **chas** - Charles River dummy variable (1 if tract bounds river, 0 otherwise)
- **nox** - Nitrogen oxide concentration (pollution level)
- **rm** - Average number of rooms per dwelling
- **age** - Proportion of owner-occupied units built before 1940
- **dis** - Weighted distance to employment centers
- **rad** - Accessibility to radial highways
- **tax** - Property tax rate per $10,000
- **ptratio** - Pupil-teacher ratio by town
- **black** - Proportion of the population that is black*
- **lstat** - Percentage of lower-status population
- **medv** - Median house value (target variable) in $1000s

*$B = 1000(Bk - 0.63)^2$ where $Bk$ is the proportion of black residents in the town

### Goals

There are three main goals of this analysis:
- Perform Exploratory Data Analysis (EDA)
- Identify the key features affecting housing prices
- Build a predictive model for housing prices

## Data Loading & Preprocessing

```r
# load data
library(MASS)
data("Boston")
```

```r
# check first few rows
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```r
# amount of missing vals
sum(is.na(Boston))
```

```
## [1] 0
```

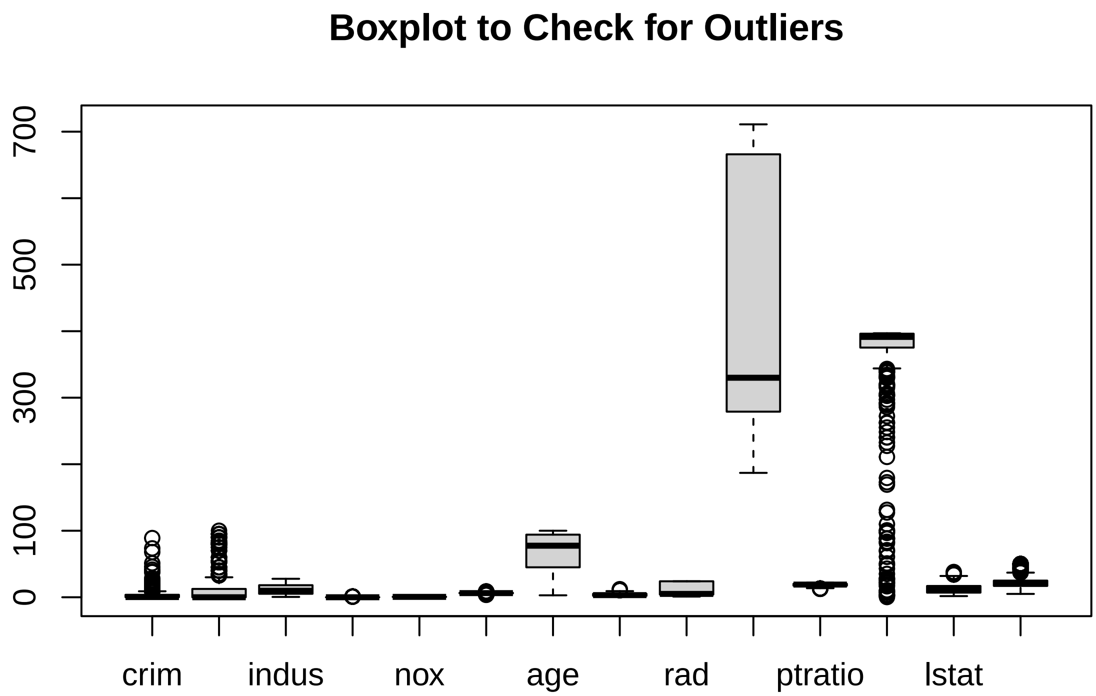
There are no missing values, so we do not need to worry about imputation, deletion, etc.

```r
# check number of duplicate rows
sum(duplicated(Boston))
```

```
## [1] 0
```

There are no duplicate rows, so we do not need to worry about row deletion here either.

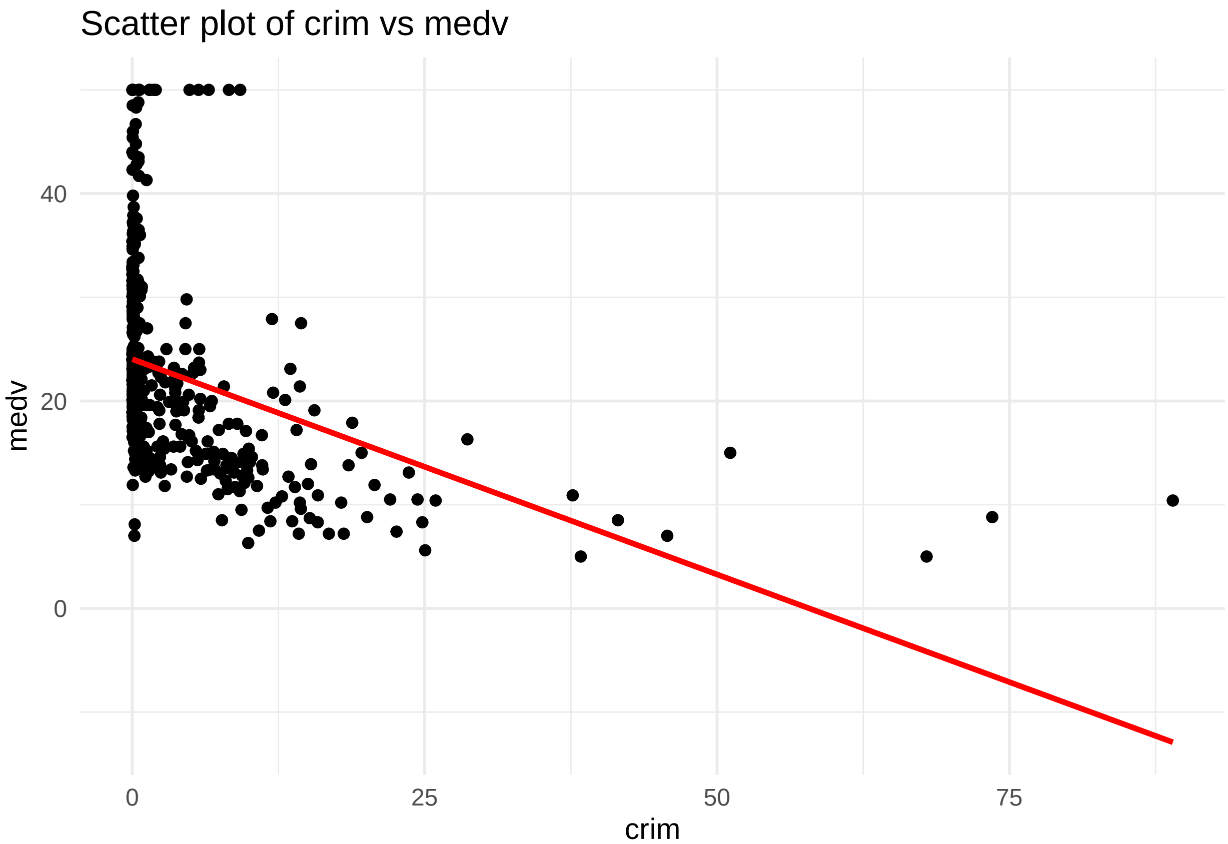
```r
boxplot(Boston, main="Boxplot to Check for Outliers")
```



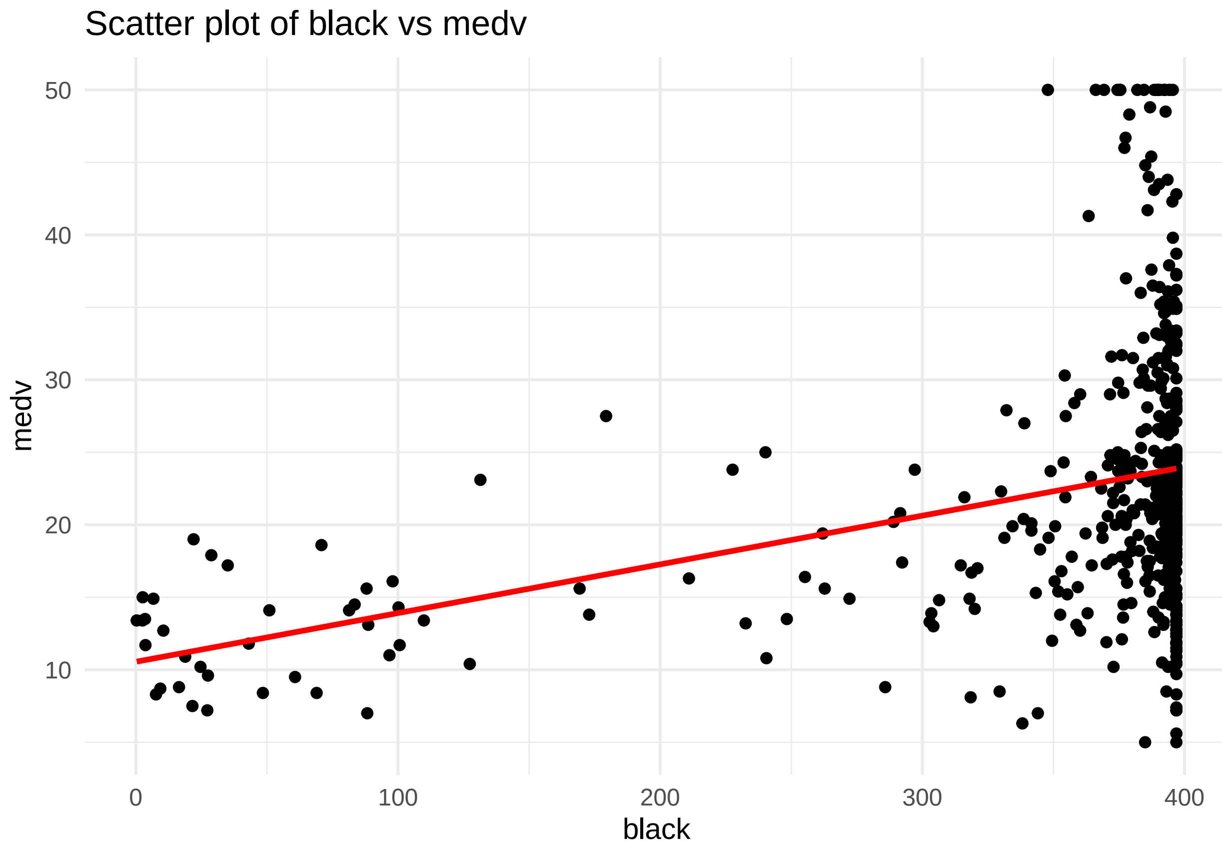

There are some outliers present, mostly in crim, zn, and black. We will check these closer with some scatter plots.

```
library(ggplot2)

independent_vars <- c("crim", "black", "zn")
dependent_var <- "medv"

# Create scatter plots for each independent variable
for (var in independent_vars) {
  p <- ggplot(Boston, aes_string(x = var, y = dependent_var)) +
    geom_point() +
    geom_smooth(method = "lm", color = "red", se = FALSE) + # add regression line for each plot
    ggtitle(paste("Scatter plot of", var, "vs", dependent_var)) +
    theme_minimal()

  print(p)
}
```

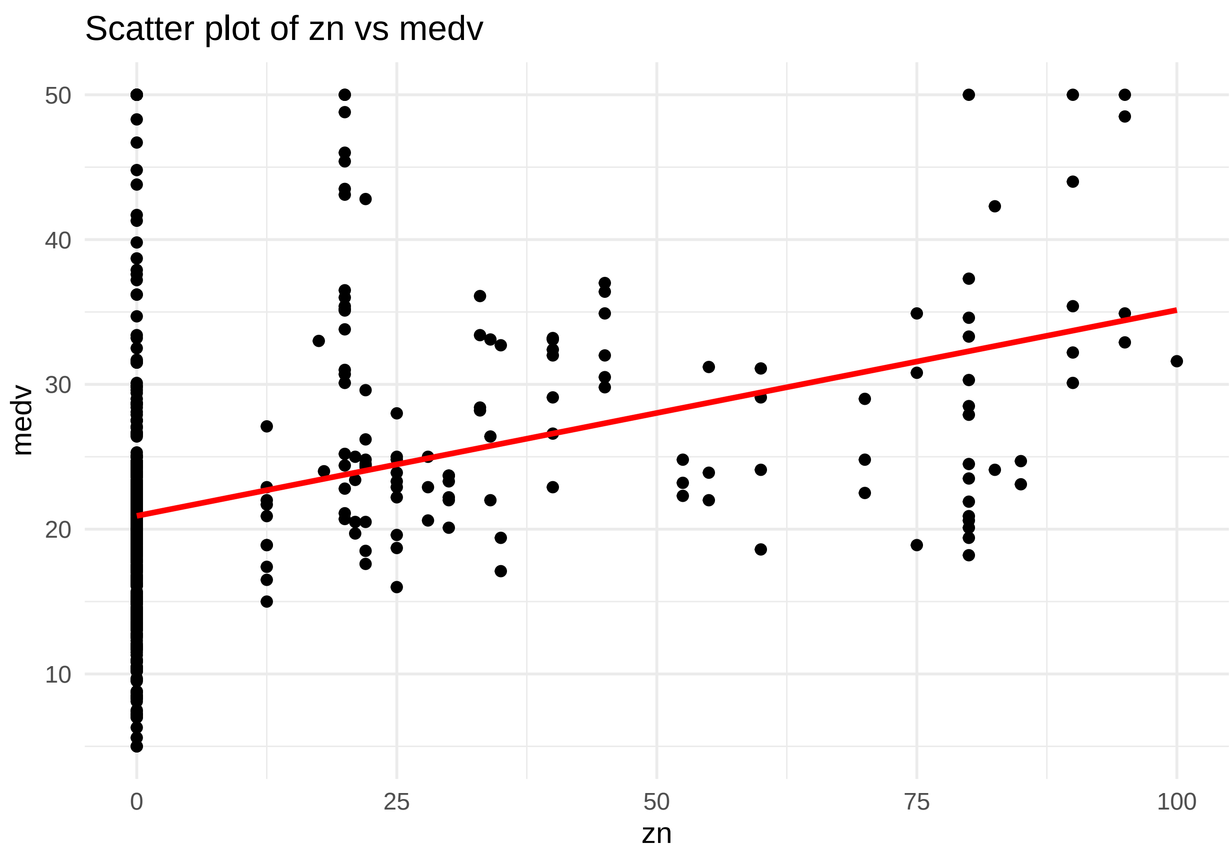
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## `geom_smooth()` using formula = 'y ~ x'
```




Now we will remove the outliers from the dataset.

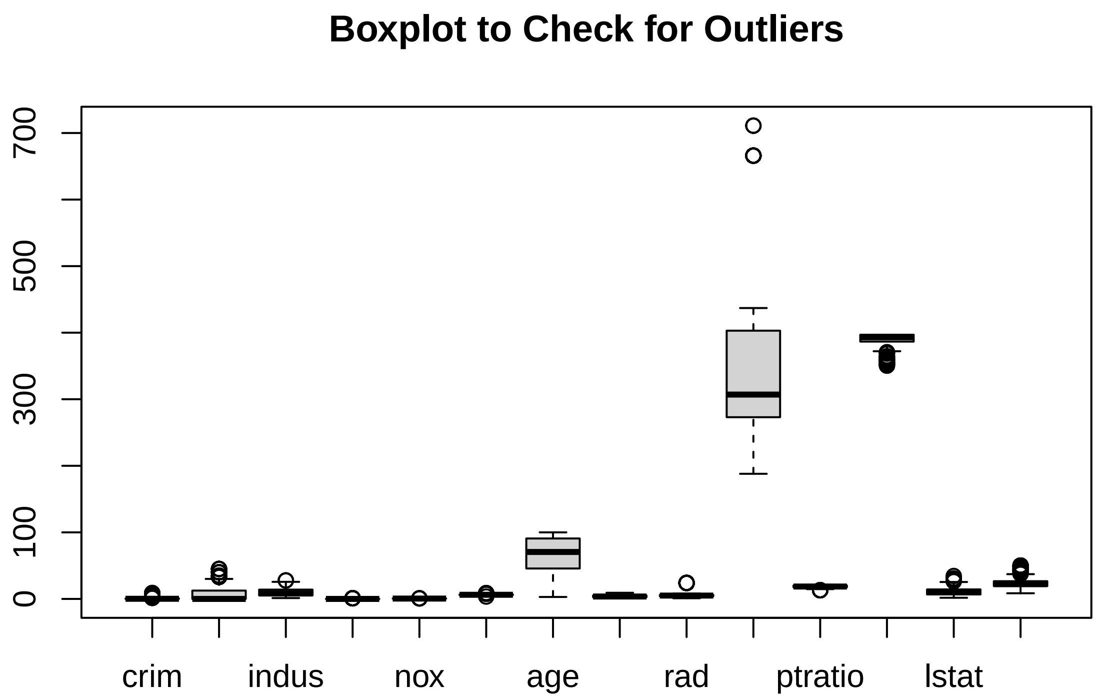
```
# remove crim outliers
Q <- quantile(Boston$crim, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(Boston$crim)

up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range

Boston<- subset(Boston, Boston$crim > (Q[1] - 1.5*iqr) & Boston$crim < (Q[2]+1.5*iqr))
```

```
# remove zn outliers
Q <- quantile(Boston$zn, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(Boston$zn)

up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range

Boston<- subset(Boston, Boston$zn > (Q[1] - 1.5*iqr) & Boston$zn < (Q[2]+1.5*iqr))
```

```
# remove black outliers
Q <- quantile(Boston$black, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(Boston$black)

up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range
```

```r
Boston<- subset(Boston, Boston$black > (Q[1] - 1.5*iqr) & Boston$black < (Q[2]+1.5*iqr))
```

```r
boxplot(Boston, main="Boxplot to Check for Outliers")
```




The box plots look much more normal now.

## Exploratory Data Analysis (EDA)

```r
dim(Boston) # check dimensions of dataset
```

```
## [1] 346  14
```

Our dataset has gone from 506 observations to 346 due to the removal of outliers.

```r
unique(Boston$chas)
```

```
## [1] 0 1
```

```r
unique(Boston$rad)
```

```
## [1]  1  2  3  5  4  8  6  7 24
```

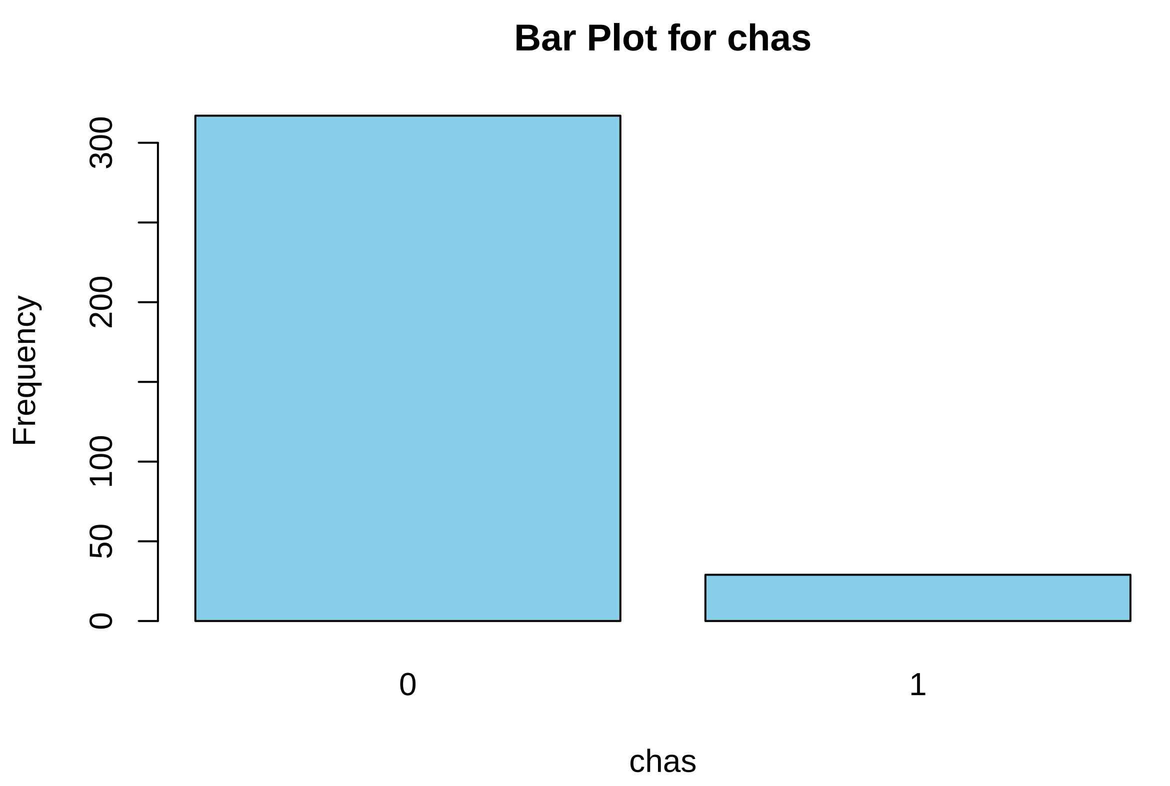
```r
table(Boston$chas)
```

```
##
##   0   1
## 317  29
```

```r
table(Boston$rad)
```

```
##
##  1  2  3  4  5  6  7  8 24
## 12 18 32 87 91 21 17 24 44
```

```r
chas_table <- table(Boston$chas)
rad_table <- table(Boston$rad)

# Create a bar plot for 'chas'
barplot(chas_table, main="Bar Plot for chas", col="skyblue", xlab="chas", ylab="Frequency")
```




```r
# Create a bar plot for 'rad'
barplot(rad_table, main="Bar Plot for rad", col="lightgreen", xlab="rad", ylab="Frequency")
```


```r
# check var types
sapply(Boston, class)
```

```
##      crim        zn     indus      chas       nox        rm       age       dis
## "numeric" "numeric" "numeric" "integer" "numeric" "numeric" "numeric" "numeric"
##       rad       tax   ptratio     black     lstat      medv
## "integer" "numeric" "numeric" "numeric" "numeric" "numeric"
```

```r
str(Boston) # display contents of dataframe
```

```
## 'data.frame':    346 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```r
summary(Boston) # summary stats for each column
```

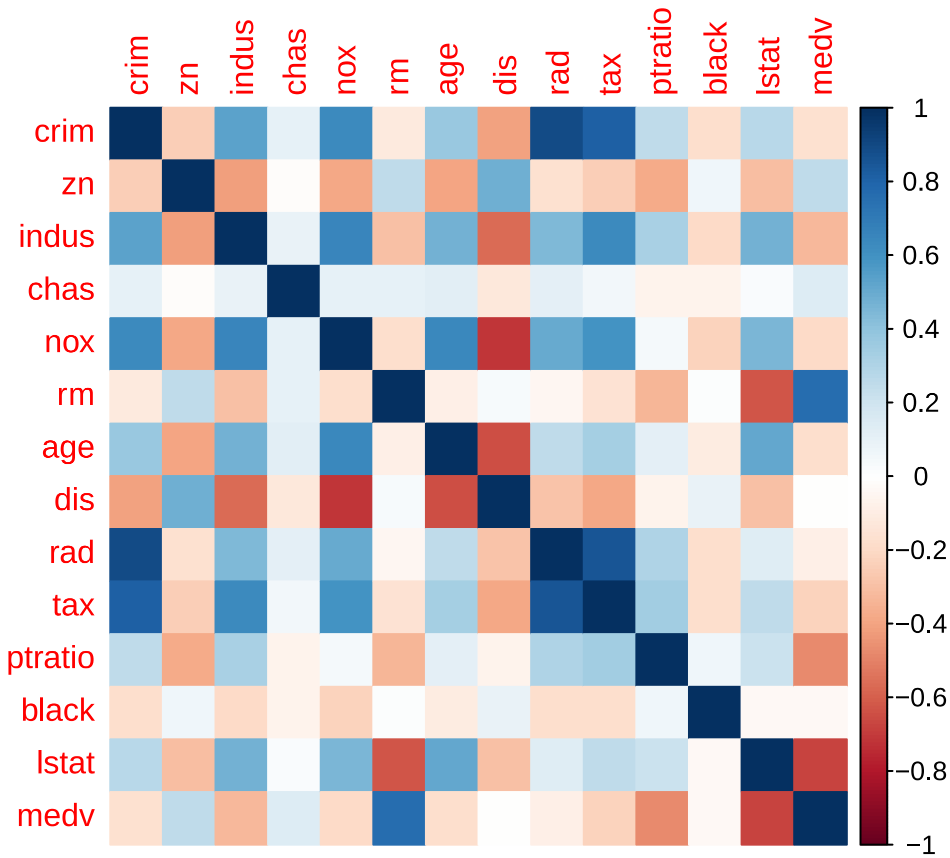
```
##       crim                zn             indus            chas        
##  Min.   :0.00632   Min.   : 0.000   Min.   : 1.25   Min.   :0.00000  
##  1st Qu.:0.08324   1st Qu.: 0.000   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median :0.17644   Median : 0.000   Median : 8.14   Median :0.00000  
##  Mean   :1.00688   Mean   : 6.611   Mean   :10.02   Mean   :0.08382  
##  3rd Qu.:0.62135   3rd Qu.:12.500   3rd Qu.:13.92   3rd Qu.:0.00000  
##  Max.   :8.98296   Max.   :45.000   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis       
##  Min.   :0.4090   Min.   :3.561   Min.   :  2.90   Min.   :1.202  
##  1st Qu.:0.4530   1st Qu.:5.934   1st Qu.: 45.73   1st Qu.:2.472  
##  Median :0.5150   Median :6.215   Median : 70.50   Median :3.609  
##  Mean   :0.5347   Mean   :6.351   Mean   : 66.36   Mean   :3.918  
##  3rd Qu.:0.5825   3rd Qu.:6.631   3rd Qu.: 90.78   3rd Qu.:5.184  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :9.223  
##       rad              tax           ptratio          black      
##  Min.   : 1.000   Min.   :188.0   Min.   :13.00   Min.   :350.4  
##  1st Qu.: 4.000   1st Qu.:273.0   1st Qu.:17.40   1st Qu.:386.6  
##  Median : 5.000   Median :307.0   Median :18.60   Median :393.2  
##  Mean   : 7.052   Mean   :356.1   Mean   :18.36   Mean   :389.1  
##  3rd Qu.: 6.000   3rd Qu.:401.8   3rd Qu.:20.20   3rd Qu.:396.9  
##  Max.   :24.000   Max.   :711.0   Max.   :21.20   Max.   :396.9  
##      lstat             medv      
##  Min.   : 1.730   Min.   : 8.50  
##  1st Qu.: 6.723   1st Qu.:19.30  
##  Median :10.040   Median :22.20  
##  Mean   :11.060   Mean   :24.21  
##  3rd Qu.:14.250   3rd Qu.:26.60  
##  Max.   :34.410   Max.   :50.00
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.95 loaded
```

```r
corrplot(cor(Boston), method = "color")
```




It seems that among the highest correlated pairs: - crim is positively correlated with rad - tax is positively correlated with crim - nox is negavitvely correlated with dis - age is negatively correlated with dis

These correlations make sense; crime is higher in more urban areas, where tax is higher; the further from urban areas, the lower the pollution rate; and there are newer living spaces in cities than in rural areas.

## Simple Linear Regression (SLR)

First, we will perform SLR using number of rooms as

```
# SLR (predict price based on just number of rooms)
model1 <- lm(medv ~ rm, data=Boston)
summary(model1)
```

```
##
## Call:
## lm(formula = medv ~ rm, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.058  -2.393  -0.429   2.183  38.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -35.2795     2.6997  -13.07   <2e-16 ***
```

```
## rm            9.3664     0.4227   22.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.344 on 344 degrees of freedom
## Multiple R-squared:  0.5881, Adjusted R-squared:  0.5869
## F-statistic: 491.1 on 1 and 344 DF,  p-value: < 2.2e-16
```

```
# check model1 performance
summary(model1)$r.squared
```

```
## [1] 0.5880782
```

# Multiple Linear Regression (MLR)

## Full Model (using all variables)

```
# MLR (predict price using all available vars)
model2 <- lm(medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lsta
summary(model2)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + chas + nox + rm + age +
##     dis + rad + tax + ptratio + black + lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.123  -2.237  -0.382   1.483  27.980
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.789e+01  1.078e+01   3.515 0.000501 ***
## crim        -1.680e-01  3.102e-01  -0.541 0.588537
## zn           1.375e-03  2.421e-02   0.057 0.954728
## indus        6.201e-02  6.049e-02   1.025 0.306036
## chas         1.920e+00  8.617e-01   2.228 0.026545 *
## nox         -1.081e+01  4.702e+00  -2.300 0.022072 *
## rm           5.215e+00  4.909e-01  10.622  < 2e-16 ***
## age          4.962e-04  1.384e-02   0.036 0.971424
## dis         -1.028e+00  2.205e-01  -4.662 4.54e-06 ***
## rad          2.614e-01  9.919e-02   2.635 0.008808 **
## tax         -1.048e-02  3.903e-03  -2.685 0.007611 **
## ptratio     -1.050e+00  1.445e-01  -7.268 2.63e-12 ***
## black       -2.747e-02  2.341e-02  -1.174 0.241427
## lstat       -5.237e-01  6.688e-02  -7.830 6.61e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.288 on 332 degrees of freedom
```

```
## Multiple R-squared:  0.744,  Adjusted R-squared:  0.7339
## F-statistic:  74.2 on 13 and 332 DF,  p-value: < 2.2e-16
```

```r
# check model2 performance
summary(model2)$adj.r.squared
```

```
## [1] 0.7339257
```

### Stepwise Selection (both ways)

```r
step_model <- step(model2, direction="both")
```

```
## Start:  AIC=1021.21
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq    RSS    AIC
## - age      1      0.02 6105.8 1019.2
## - zn       1      0.06 6105.8 1019.2
## - crim     1      5.39 6111.1 1019.5
## - indus    1     19.33 6125.1 1020.3
## - black    1     25.33 6131.1 1020.6
## <none>                 6105.7 1021.2
## - chas     1     91.30 6197.0 1024.3
## - nox      1     97.28 6203.0 1024.7
## - rad      1    127.69 6233.4 1026.4
## - tax      1    132.61 6238.4 1026.6
## - dis      1    399.76 6505.5 1041.2
## - ptratio  1    971.52 7077.3 1070.3
## - lstat    1   1127.57 7233.3 1077.8
## - rm       1   2075.08 8180.8 1120.4
##
## Step:  AIC=1019.21
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq    RSS    AIC
## - zn       1      0.05 6105.8 1017.2
## - crim     1      5.39 6111.2 1017.5
## - indus    1     19.33 6125.1 1018.3
## - black    1     25.40 6131.2 1018.6
## <none>                 6105.8 1019.2
## + age      1      0.02 6105.7 1021.2
## - chas     1     91.63 6197.4 1022.4
## - nox      1    101.32 6207.1 1022.9
## - rad      1    127.74 6233.5 1024.4
## - tax      1    132.59 6238.4 1024.6
## - dis      1    448.07 6553.8 1041.7
## - ptratio  1    984.34 7090.1 1068.9
## - lstat    1   1399.05 7504.8 1088.6
## - rm       1   2295.99 8401.8 1127.7
```

```
##
## Step:  AIC=1017.21
## medv ~ crim + indus + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq    RSS    AIC
## - crim     1      5.42 6111.2 1015.5
## - indus    1     19.27 6125.1 1016.3
## - black    1     25.35 6131.2 1016.6
## <none>                 6105.8 1017.2
## + zn       1      0.05 6105.8 1019.2
## + age      1      0.02 6105.8 1019.2
## - chas     1     91.77 6197.6 1020.4
## - nox      1    101.74 6207.6 1020.9
## - rad      1    128.04 6233.9 1022.4
## - tax      1    132.88 6238.7 1022.7
## - dis      1    497.03 6602.9 1042.3
## - ptratio  1   1113.15 7219.0 1073.2
## - lstat    1   1399.46 7505.3 1086.6
## - rm       1   2317.23 8423.1 1126.5
##
## Step:  AIC=1015.52
## medv ~ indus + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq    RSS    AIC
## - indus    1     18.39 6129.6 1014.6
## - black    1     26.08 6137.3 1015.0
## <none>                 6111.2 1015.5
## + crim     1      5.42 6105.8 1017.2
## + zn       1      0.09 6111.2 1017.5
## + age      1      0.01 6111.2 1017.5
## - chas     1     92.77 6204.0 1018.7
## - nox      1    123.03 6234.3 1020.4
## - tax      1    133.55 6244.8 1021.0
## - rad      1    178.79 6290.0 1023.5
## - dis      1    497.75 6609.0 1040.6
## - ptratio  1   1110.86 7222.1 1071.3
## - lstat    1   1441.29 7552.5 1086.8
## - rm       1   2313.84 8425.1 1124.6
##
## Step:  AIC=1014.56
## medv ~ chas + nox + rm + dis + rad + tax + ptratio + black +
##     lstat
##
##           Df Sum of Sq    RSS    AIC
## - black    1     32.10 6161.7 1014.4
## <none>                 6129.6 1014.6
## + indus    1     18.39 6111.2 1015.5
## + crim     1      4.54 6125.1 1016.3
## + age      1      0.03 6129.6 1016.6
## + zn       1      0.00 6129.6 1016.6
## - chas     1    101.20 6230.8 1018.2
## - nox      1    108.48 6238.1 1018.6
```

```
## - tax      1    115.23 6244.9 1019.0
## - rad      1    161.66 6291.3 1021.6
## - dis      1    564.50 6694.1 1043.0
## - ptratio  1   1103.21 7232.8 1069.8
## - lstat    1   1425.68 7555.3 1084.9
## - rm       1   2299.01 8428.6 1122.8
##
## Step:  AIC=1014.37
## medv ~ chas + nox + rm + dis + rad + tax + ptratio + lstat
##
##           Df Sum of Sq    RSS    AIC
## <none>                 6161.7 1014.4
## + black    1     32.10 6129.6 1014.6
## + indus    1     24.40 6137.3 1015.0
## + crim     1      5.15 6156.6 1016.1
## + zn       1      0.07 6161.7 1016.4
## + age      1      0.00 6161.7 1016.4
## - nox      1     92.88 6254.6 1017.5
## - chas     1    105.93 6267.7 1018.3
## - tax      1    112.65 6274.4 1018.6
## - rad      1    166.37 6328.1 1021.6
## - dis      1    546.25 6708.0 1041.8
## - ptratio  1   1142.39 7304.1 1071.2
## - lstat    1   1455.81 7617.5 1085.8
## - rm       1   2284.20 8445.9 1121.5
```

```r
summary(step_model)
```

```
##
## Call:
## lm(formula = medv ~ chas + nox + rm + dis + rad + tax + ptratio +
##     lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0335  -2.1727  -0.4125   1.6477  28.3458
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.848692   5.478995   4.900 1.49e-06 ***
## chas         2.054760   0.853684   2.407  0.01662 *
## nox         -9.590556   4.255198  -2.254  0.02485 *
## rm           5.175368   0.463032  11.177  < 2e-16 ***
## dis         -1.048531   0.191833  -5.466 8.98e-08 ***
## rad          0.210108   0.069653   3.017  0.00275 **
## tax         -0.008859   0.003569  -2.482  0.01355 *
## ptratio     -1.035078   0.130949  -7.904 3.87e-14 ***
## lstat       -0.521793   0.058477  -8.923  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.276 on 337 degrees of freedom
## Multiple R-squared:  0.7416, Adjusted R-squared:  0.7355
## F-statistic: 120.9 on 8 and 337 DF,  p-value: < 2.2e-16
```

## Random Forest Variable Importance

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
rf_model <- randomForest(medv ~ ., data=Boston, importance=TRUE)
importance(rf_model)
```

```
##            %IncMSE IncNodePurity
## crim     9.6097746     754.75526
## zn       6.2691136     290.68763
## indus   15.0494830    1609.08597
## chas     0.7354297      96.85275
## nox     12.0641031     775.85107
## rm      32.1211948    8060.82280
## age     13.2464350    1046.44294
## dis     12.4774555    1333.48326
## rad      5.6943039     219.22666
## tax     14.0549720     608.52364
## ptratio 15.3123393    1443.25744
## black    5.7657150     500.86099
## lstat   29.8055202    6768.56628
```

```r
giga_rf_model <- lm(medv ~ rm + lstat + indus + ptratio + nox + tax + crim, data = Boston)
summary(giga_rf_model)
```

```
##
## Call:
## lm(formula = medv ~ rm + lstat + indus + ptratio + nox + tax +
##     crim, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.766  -2.423  -0.557   1.520  30.852
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.861016   5.121037   2.316   0.0211 *
```

```
## rm           5.786486   0.478989  12.081  < 2e-16 ***
## lstat       -0.544141   0.062193  -8.749  < 2e-16 ***
## indus        0.115969   0.059852   1.938   0.0535 .
## ptratio     -0.997903   0.139698  -7.143 5.63e-12 ***
## nox          1.302475   4.020190   0.324   0.7462
## tax         -0.006473   0.003371  -1.920   0.0557 .
## crim         0.377774   0.235082   1.607   0.1090
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.508 on 338 degrees of freedom
## Multiple R-squared:  0.7119, Adjusted R-squared:  0.706
## F-statistic: 119.3 on 7 and 338 DF,  p-value: < 2.2e-16
```

### Lasso Regression

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

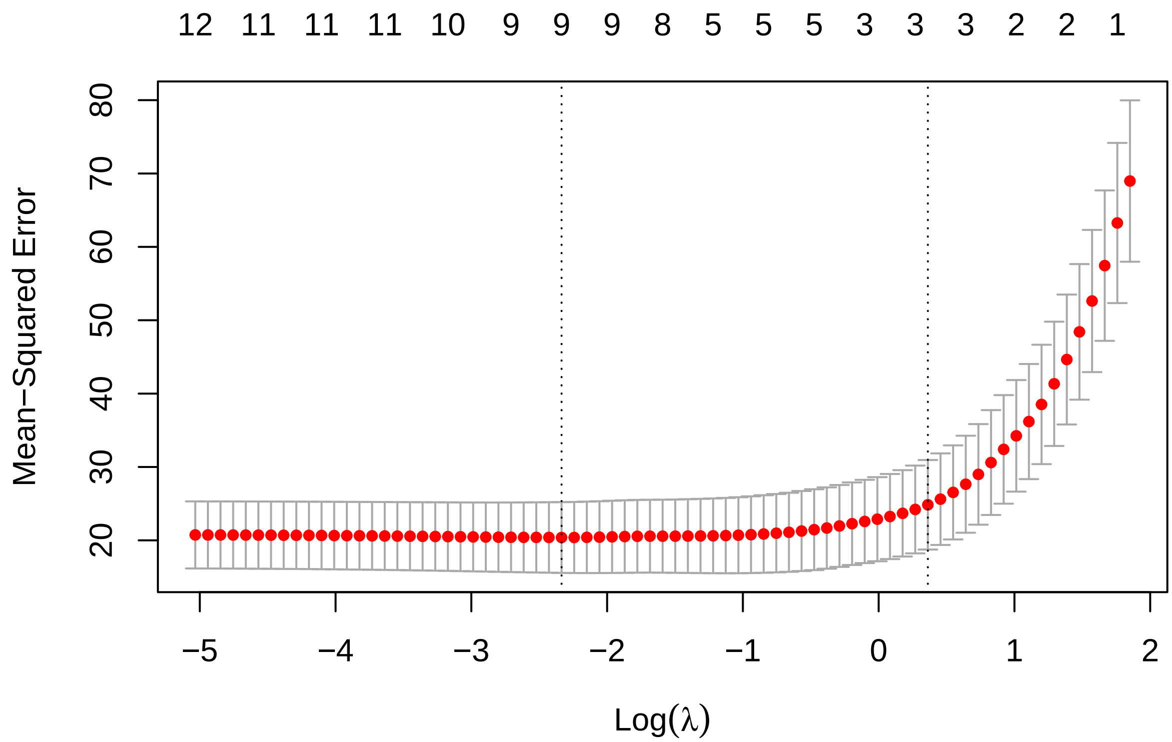
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
x <- model.matrix(medv ~ ., Boston)[, -1]
y <- Boston$medv

cv_model <- cv.glmnet(x, y, alpha = 1)

best_lambda <- cv_model$lambda.min
best_lambda
```

```
## [1] 0.09676229
```

```r
plot(cv_model)
```




```r
best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
coef(best_model)
```

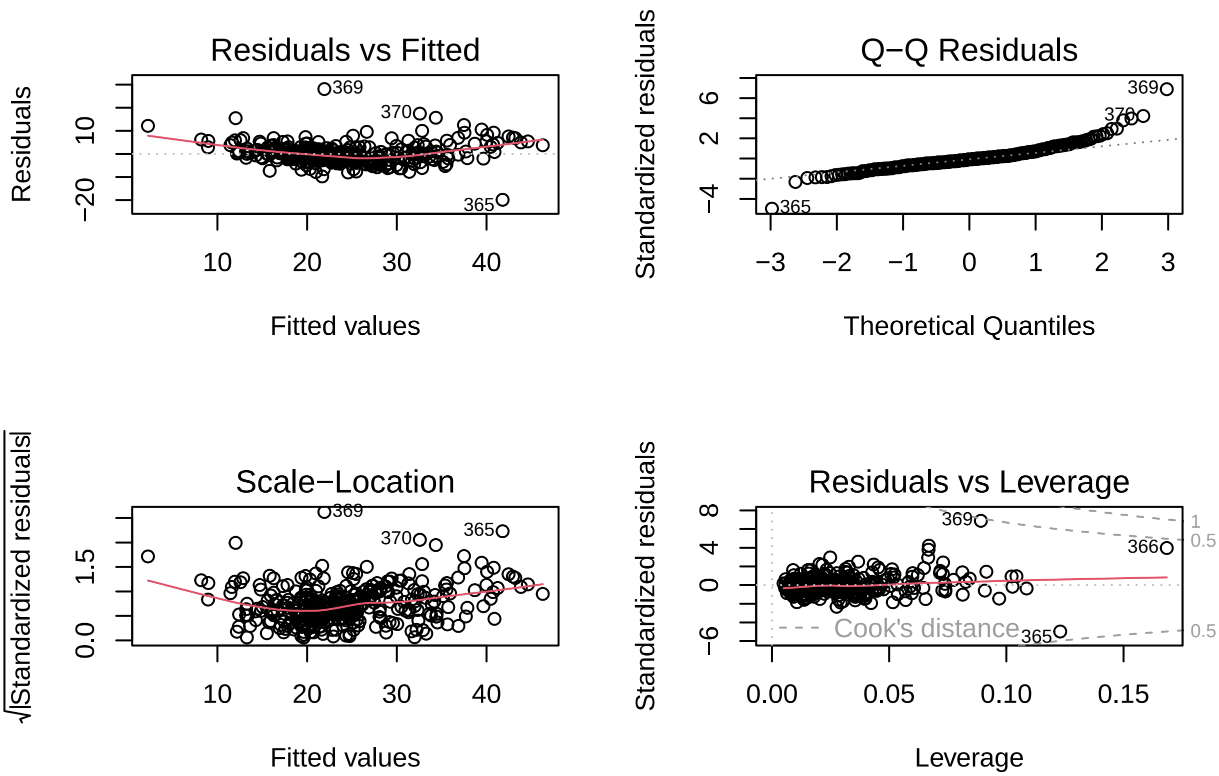
```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 28.268703559
## crim         .
## zn           .
## indus        .
## chas         1.931158865
## nox         -5.738957873
## rm           5.274145970
## age          .
## dis         -0.820565103
## rad          0.094852800
## tax         -0.004451904
## ptratio     -0.958710332
## black       -0.018327637
## lstat       -0.523374404
```

```r
lasso_model <- lm(medv ~ chas + nox + rm + dis + rad + tax + ptratio + black + lstat, data = Boston)
summary(lasso_model)
```

```
##
## Call:
```

```
## lm(formula = medv ~ chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8917  -2.1563  -0.3709   1.5052  28.0770
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  38.955123  10.642073   3.660 0.000292 ***
## chas          2.010002   0.853391   2.355 0.019082 *
## nox         -10.498027   4.305118  -2.438 0.015266 *
## rm            5.194699   0.462740  11.226  < 2e-16 ***
## dis          -1.069536   0.192270  -5.563 5.43e-08 ***
## rad           0.207216   0.069609   2.977 0.003123 **
## tax          -0.008962   0.003566  -2.513 0.012429 *
## ptratio      -1.020674   0.131252  -7.776 9.23e-14 ***
## black        -0.030626   0.023089  -1.326 0.185594
## lstat        -0.517252   0.058511  -8.840  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.271 on 336 degrees of freedom
## Multiple R-squared:  0.7429, Adjusted R-squared:  0.7361
## F-statistic: 107.9 on 9 and 336 DF,  p-value: < 2.2e-16
```

### Checking Assumptions




The diagnostic plots overall look okay.
- Residuals vs. Fitted shows random scatter of points around horizontal line at 0, which suggests linearity.
- QQ Residuals points look to lie on the diagonal line, suggesting normally distributed residuals (some slight deviations at the ends).
- Scale-Location shows a random spread of points, but the line is not horizontal, suggesting issues with heteroscedasticity.
- Residuals vs. Leverage looks overall good, with the points clustering towards the middle line, suggesting few unduly influential observations.

```r
# check independence of errors
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.3.3

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
dwtest(lasso_model)
```

```
##
##  Durbin-Watson test
##
## data:  lasso_model
## DW = 1.1944, p-value = 1.909e-15
## alternative hypothesis: true autocorrelation is greater than 0
```

The Durbin-Watson test suggests that the residuals are autocorrelated.

```r
bptest(lasso_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lasso_model
## BP = 54.687, df = 9, p-value = 1.397e-08
```

The Breusch-Pagan test suggests that the residuals violate the homoscedasticity assumption.

```r
shapiro.test(lasso_model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lasso_model$residuals
## W = 0.89708, p-value = 1.514e-14
```

The Shapiro-Wilk test suggests that the residuals violate the normality assumption.

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```r
vif(lasso_model)
```

```
##     chas      nox       rm      dis      rad      tax  ptratio    black
## 1.060674 3.432122 1.876214 2.203602 4.063986 4.643144 1.381531 1.082753
##    lstat
## 2.112294
```

The Variance Inflation Factor does not indicate any multicollinearity among the variables.

# Conclusion and Insights

## Key Findings

It seems that rm, ptratio, and lstat are consistently the most important variables when it comes to predicting house price. They were selected as significant and influential by all of the models created in this analysis.

The model with the best predictive power was the lasso regression model, with RSE of 4.271 and adjusted $R^2$ of 0.7361. The model is as follows:

$medv = 39 + 2chas - 10.5nox + 5.2rm - dis + 0.2rad - ptratio - 0.5lstat$

However, the performance metrics of all models were close; all had RSE of around 4.2 and Adjusted $R^2$ of 0.7. The lasso regression model's metrics were just barely above the others'.

What this model means, on average:
- Having a river on the tract of land (or on its boundary) corresponds to a higher house price.
- The higher the nitrogen oxide concentration in the house's area, the lower the price of the house.
- The more rooms in a house, the higher the price is.
- The further the house from an employment center, the lower the price of the house.
- The higher the radial highway accessibility, the higher the price of the house.
- The higher pupil-teacher ratio of the area, the lower the price.
- The higher the percentage of low-status population, the lower the price of the house.

These findings are in accordance with common sense. People generally like to have scenic views of a waterway, more rooms, less pollution, less commute to work and highways, better schools, and less people of low status.

Surprisingly, crime was shown to have little effect on the models, suggesting that crime and house price are not closely related.

# Potential Improvements

Removing outliers from the dataset shrunk the amount of observations to train a regression on, which can impact its representativeness to the general Boston housing market. In future studies, other methods of determining outliers can be employed to perhaps limit the amount of data loss.

The variable chas is very unbalanced, with 317 "0"s and only 29 "1"s (317 properties without rivers, 29 with). Future studies should attempt oversampling techniques to balance this variable.

Further, several assumptions of linear regression were violated. For future studies, applying weighted least squares regressions or transformations could yield a model without homoscedasticity or normality violations; generalized least squares could be employed to handle autocorrelation; and robust regression techniques can improve normality assumption violations.

Other, non-parametric models should also be considered.

Finally, this dataset is rather old (from around 1978) and is most likely unrepresentative of the current housing market trends of Boston. For relevant trends, more rencently compiled datasets should be examined.