Predicting Boston House Prices

Linear Regression

```r
# Load Boston Dataset
dim(Boston)
```

```
## [1] 506  14
```

```r
boston <- as.data.frame(Boston) #creating boston dataset
```

1. Describe the data and variables that are part of the `Boston` dataset. Tidy data as necessary.

Solution: In the Boston dataset, there are 506 rows and 14 columns. There are no NA or duplicated values in the data set. The description of the columns of the dataset is as follows:

crim: per capita crime rate by town.

zn:proportion of residential land zoned for lots over 25,000 sq.ft.

indus:proportion of non-retail business acres per town.

chas:Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).

nox:nitrogen 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 mean of 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.

black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.

lstat: lower status of the population (percent).

medv: median value of owner-occupied homes in $1000s.

```r
summary(boston) #looking at the statistics
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
```

```
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
view(boston)
dim(boston)   #dimenstions of dataset
```

```
## [1] 506  14
```

```r
sum(is.na(boston)) # checking for NA values
```

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

```r
sum(duplicated(boston)) #checking for duplicated values
```

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

2. Consider this data in context, what is the response variable of interest?

Solution: In this question, we have medv which is median value of owner-occupied homes in $1000 and the dataset contains information about median house value for 506 neighborhoods in Boston, M. So, we can take it as our response variale here and keep other variables as predictor variables.

3. For each predictor, fit a simple linear regression model to predict the response. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

Solution: We computed fit and residual for each of the predictor variables.

```
 1. for Zn we have p value as '2.2e-16 ***' and hence it is statistically significantly
 2. for crim we have p value as '2.2e-16 ***' and hence it is statistically significantly
 3. for indus we have p value as '2.2e-16 ***' and hence it is statistically significantly
 4. for chas we have p value as ' 7.391e-05 ***' and hence it is statistically significantly
 5. for nox we have p value as '2.2e-16 ***' and hence it is statistically significantly
 6. for rm we have p value as '2.2e-16 ***' and hence it is statistically significantly
 7. for age we have p value as '2.2e-16 ***' and hence it is statistically significantly
 8. for dis we have p value as ' 1.207e-08 ***' and hence it is statistically significantly
 9. for rad we have p value as '2.2e-16 ***' and hence it is statistically significantly
10. for tax we have p value as '2.2e-16 ***' and hence it is statistically significantly
11. for ptratio we have p value as '2.2e-16 ***' and hence it is statistically significantly
12. for black we have p value as ' 1.318e-14 ***' and hence it is statistically significantly
13.  for lstat we have p value as '2.2e-16 ***' and hence it is statistically significantly
```

In all the predictor variables we have statistically significant association between the predictor and the response variable. We have made the plots as follows:

```r
#1.zn
fit_zn <- lm(medv ~ zn, boston)
summary(fit_zn)
```

```
##
## Call:
## lm(formula = medv ~ zn, data = boston)
##
## Residuals:
```

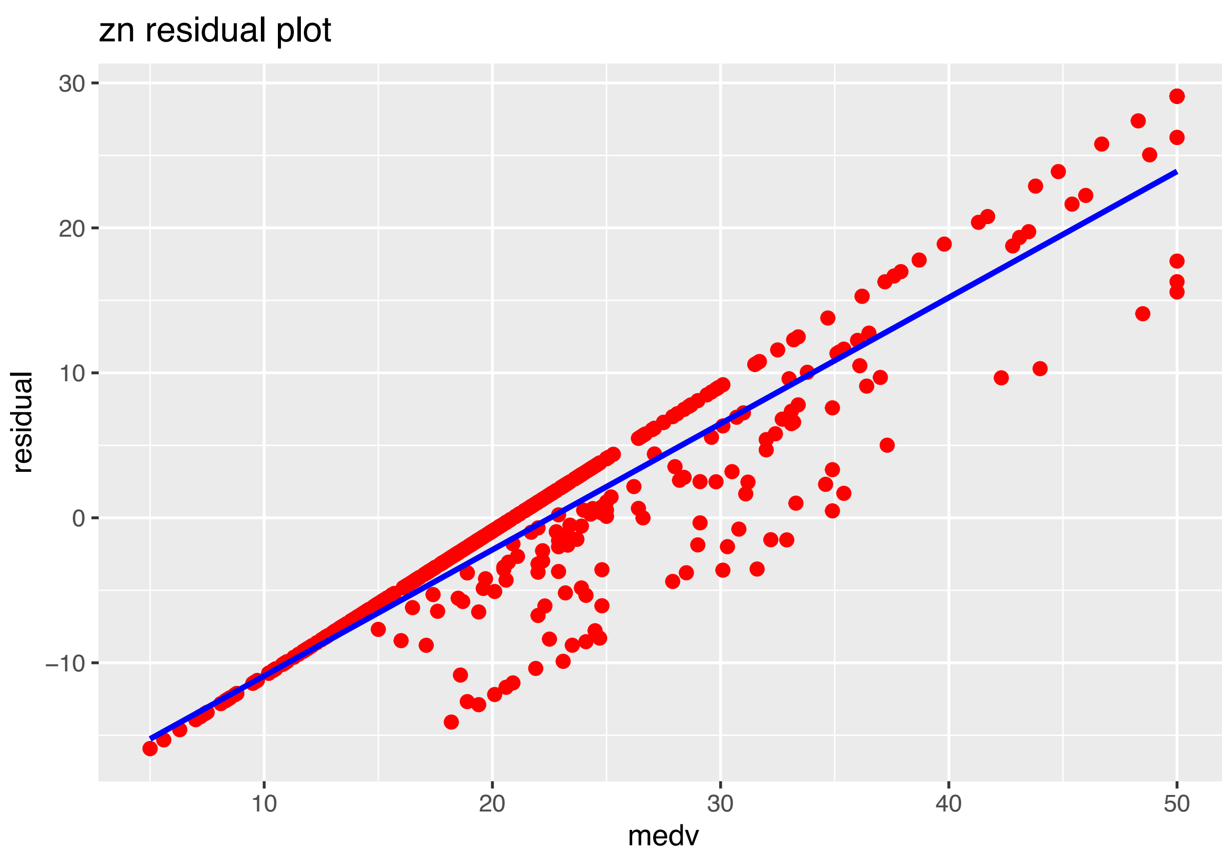
```
##      Min       1Q   Median       3Q      Max
## -15.918   -5.518   -1.006    2.757   29.082
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.91758    0.42474  49.248   <2e-16 ***
## zn           0.14214    0.01638   8.675   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.587 on 504 degrees of freedom
## Multiple R-squared:  0.1299, Adjusted R-squared:  0.1282
## F-statistic: 75.26 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
confint(fit_zn, 'zn', level = 0.95)
```

```
##        2.5 %    97.5 %
## zn 0.1099491 0.1743309
```

```r
residuals_zn <- resid(fit_zn)
plotResiduals_zn <- ggplot(data = data.frame(x = boston$medv, y= residuals_zn), aes(x = x, y=y)) + geom_

plotResiduals_zn <- plotResiduals_zn +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "zn residual plot", y = 'residual'
plotResiduals_zn
```




```r
#2 indus
fit_indus <- lm(medv ~ indus, boston)
```

```r
summary(fit_indus)
```

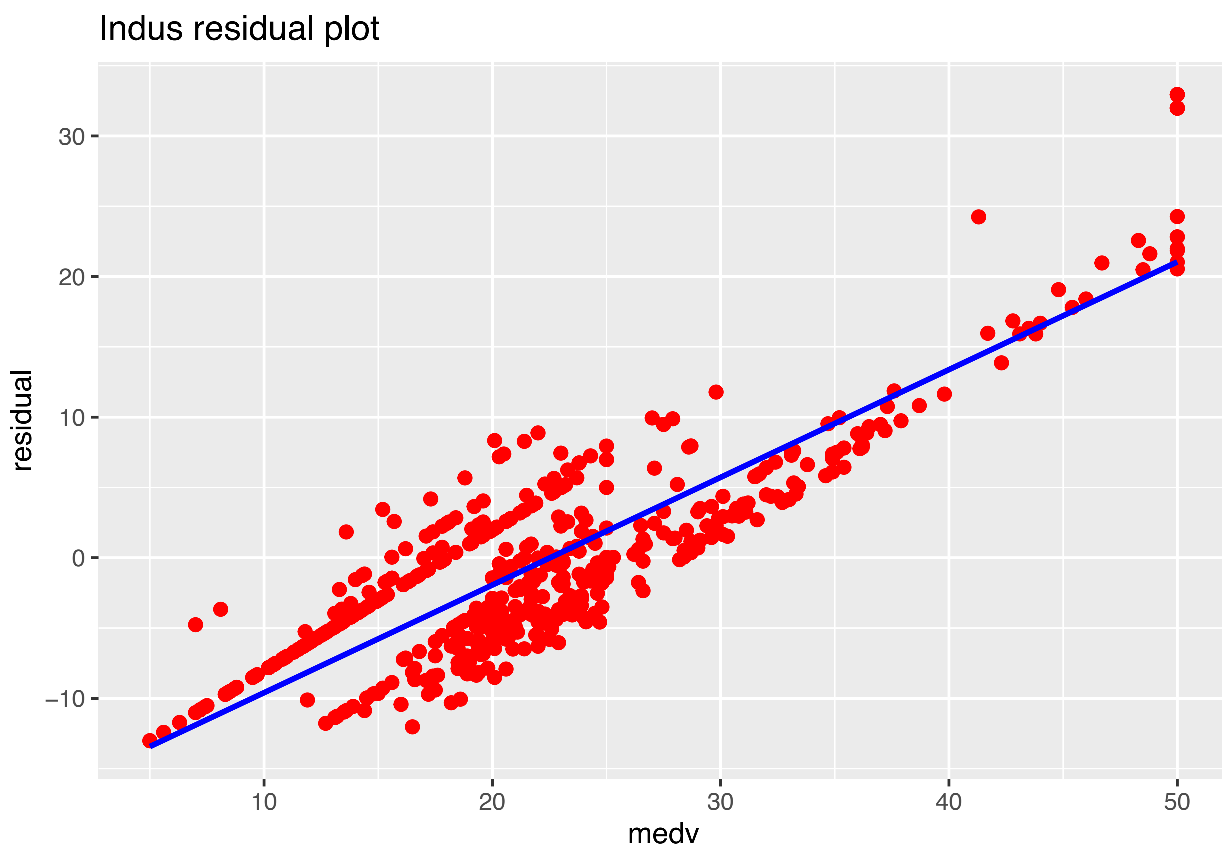
```
##
## Call:
## lm(formula = medv ~ indus, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.017  -4.917  -1.457   3.180  32.943
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.75490    0.68345   43.54   <2e-16 ***
## indus       -0.64849    0.05226  -12.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.234,  Adjusted R-squared:  0.2325
## F-statistic:   154 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
confint(fit_indus, 'indus', level = 0.95)
```

```
##            2.5 %    97.5 %
## indus -0.7511731 -0.545807
```

```r
residuals_indus <- resid(fit_indus)
plotResiduals_indus <- ggplot(data = data.frame(x = boston$medv, y= residuals_indus), aes(x = x, y=y)) +

plotResiduals_indus <- plotResiduals_indus +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Indus residual plot", y = 'residua
plotResiduals_indus
```




```
#3 boston
fit_chas <- lm(medv ~ chas, boston)
summary(fit_chas)
```

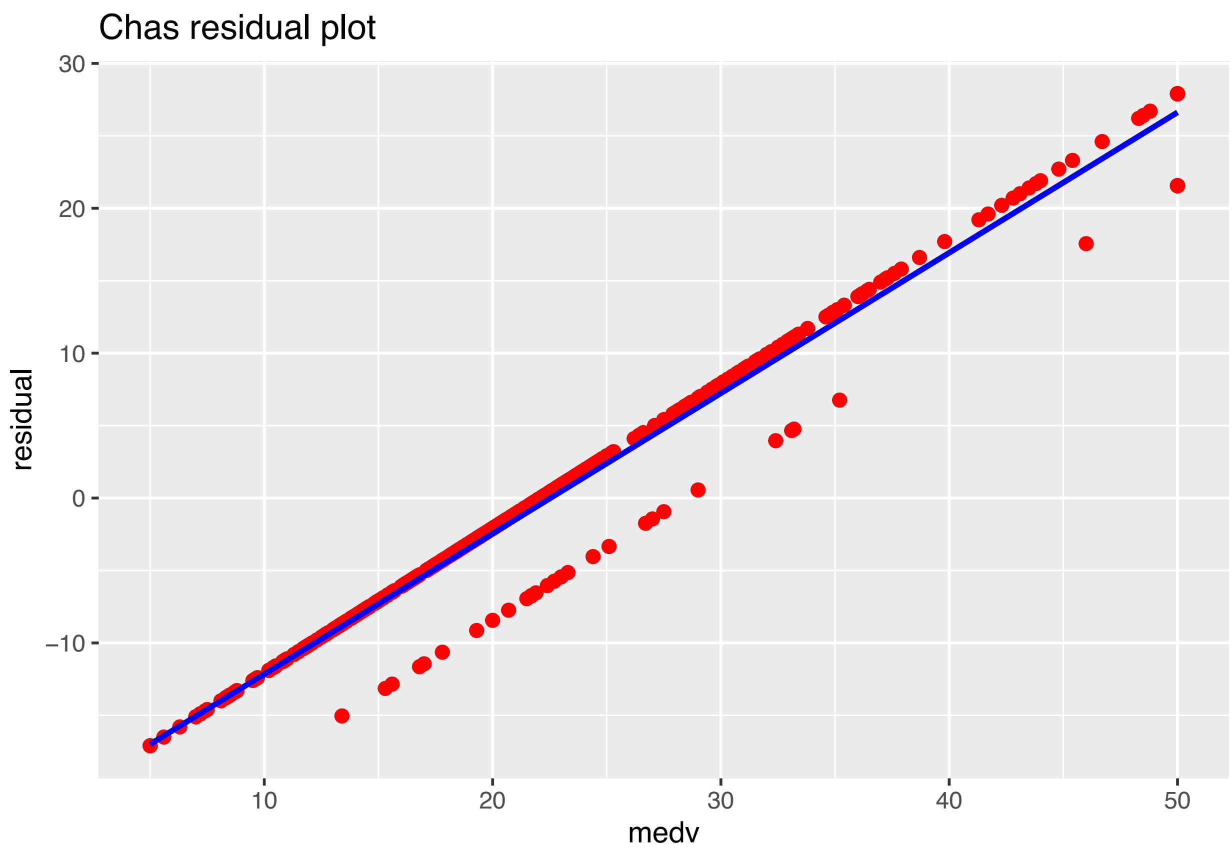
```
##
## Call:
## lm(formula = medv ~ chas, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.094  -5.894  -1.417   2.856  27.906
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.0938     0.4176  52.902  < 2e-16 ***
## chas          6.3462     1.5880   3.996 7.39e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.064 on 504 degrees of freedom
## Multiple R-squared:  0.03072,    Adjusted R-squared:  0.02879
## F-statistic: 15.97 on 1 and 504 DF,  p-value: 7.391e-05
```

```
confint(fit_chas, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_chas <- resid(fit_chas)
plotResiduals_chas <- ggplot(data = data.frame(x = boston$medv, y= residuals_chas), aes(x = x, y=y)) + g

plotResiduals_chas <- plotResiduals_chas +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Chas residual plot", y = 'residual
plotResiduals_chas
```




```r
#4
fit_nox <- lm(medv ~ nox, boston)
summary(fit_nox)
```

```
##
## Call:
## lm(formula = medv ~ nox, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.691  -5.121  -2.161   2.959  31.310
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.346      1.811   22.83   <2e-16 ***
## nox          -33.916      3.196  -10.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

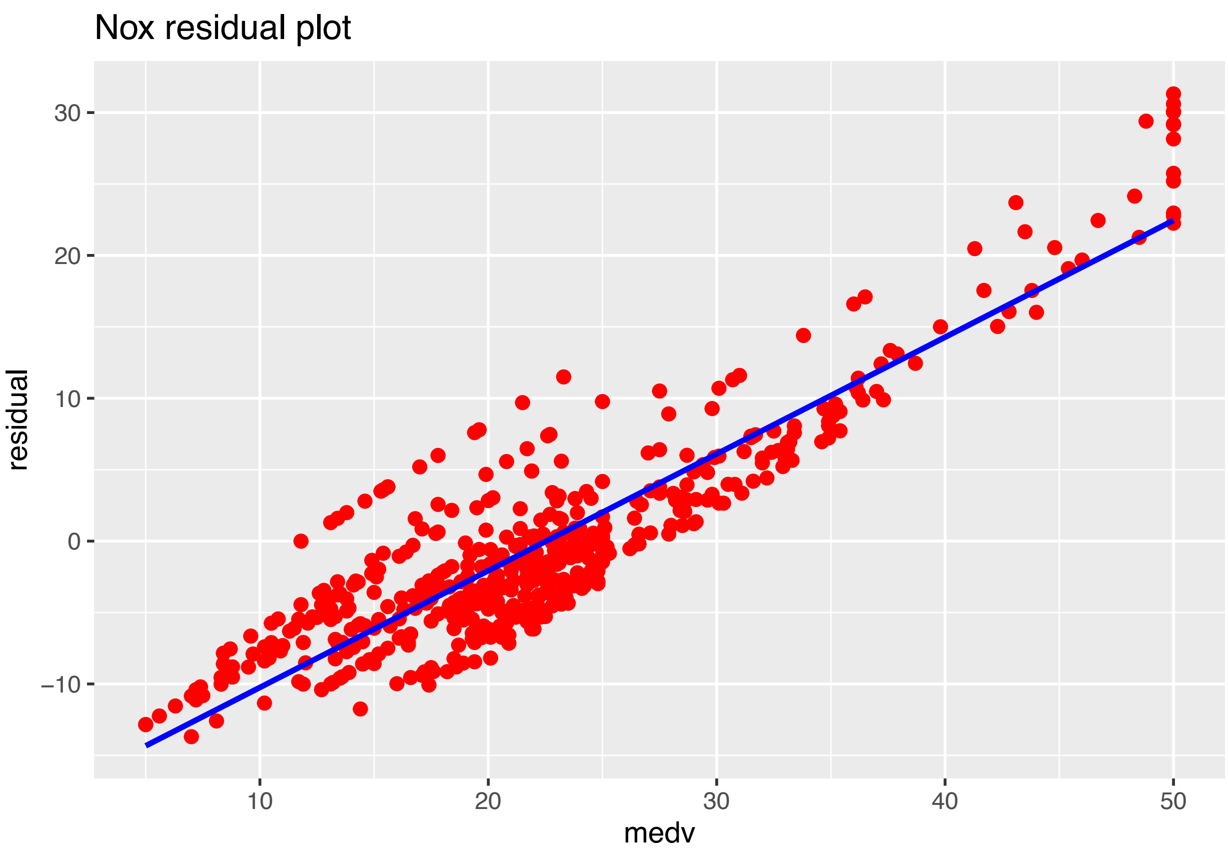
```
## Residual standard error: 8.323 on 504 degrees of freedom
## Multiple R-squared:  0.1826, Adjusted R-squared:  0.181
## F-statistic: 112.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
confint(fit_nox, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```
residuals_nox <- resid(fit_nox)
plotResiduals_nox <- ggplot(data = data.frame(x = boston$medv, y= residuals_nox), aes(x = x, y=y)) + ge

plotResiduals_nox <- plotResiduals_nox +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Nox residual plot", y = 'residual
plotResiduals_nox
```




```
#5
fit_rm <- lm(medv ~ rm, boston)
summary(fit_rm)
```

```
##
## Call:
## lm(formula = medv ~ rm, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.346  -2.547   0.090   2.986  39.433
##
```

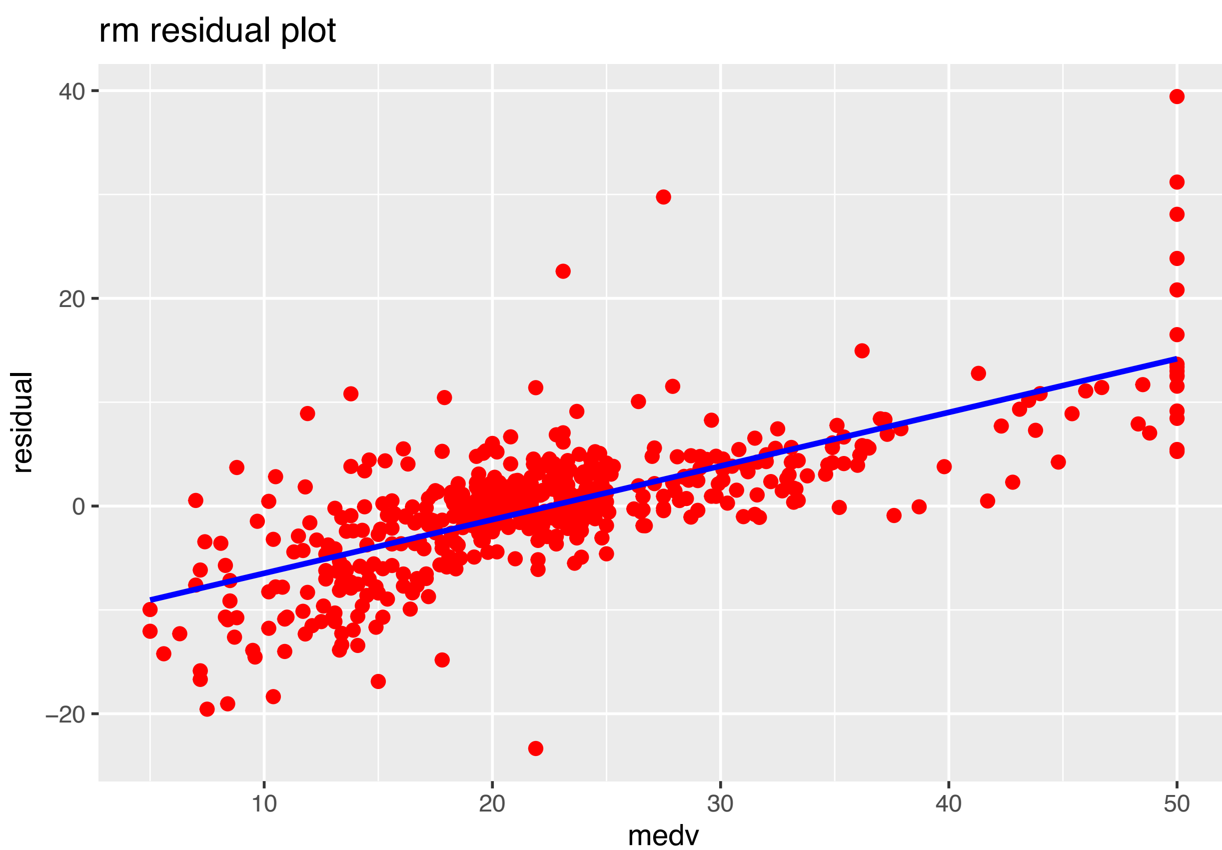
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -34.671      2.650  -13.08   <2e-16 ***
## rm             9.102      0.419   21.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.616 on 504 degrees of freedom
## Multiple R-squared:  0.4835, Adjusted R-squared:  0.4825
## F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
confint(fit_rm, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_rm <- resid(fit_rm)
plotResiduals_rm <- ggplot(data = data.frame(x = boston$medv, y= residuals_rm), aes(x = x, y=y)) + geom_

plotResiduals_rm <- plotResiduals_rm +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "rm residual plot", y = 'residual'
plotResiduals_rm
```




```r
#6
fit_age<- lm(medv ~ age, boston)
summary(fit_age)
```

```
##
```

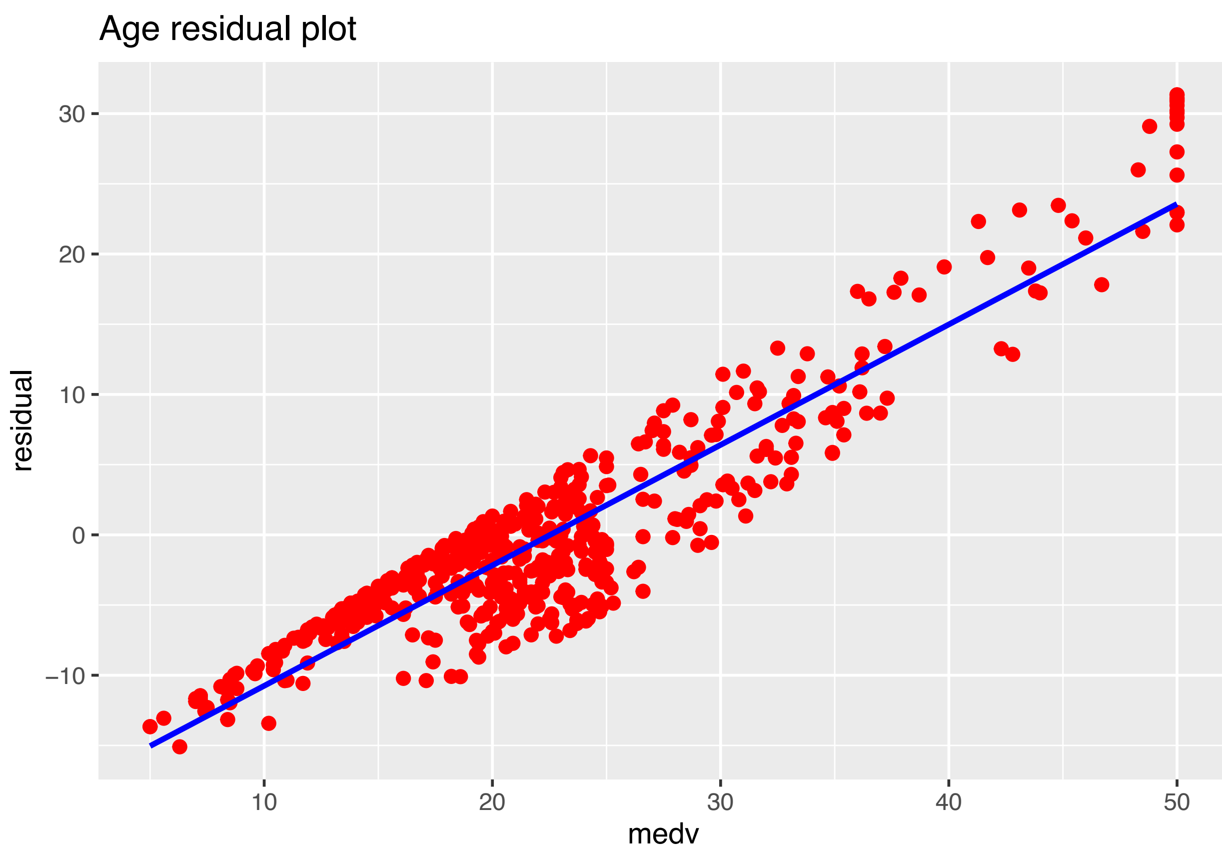
```
## Call:
## lm(formula = medv ~ age, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.097  -5.138  -1.958   2.397  31.338
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.97868    0.99911  31.006   <2e-16 ***
## age         -0.12316    0.01348  -9.137   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.527 on 504 degrees of freedom
## Multiple R-squared:  0.1421, Adjusted R-squared:  0.1404
## F-statistic: 83.48 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
confint(fit_age, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_age <- resid(fit_age)
plotResiduals_age <- ggplot(data = data.frame(x = boston$medv, y= residuals_age), aes(x = x, y=y)) + ge

plotResiduals_age <- plotResiduals_age +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Age residual plot", y = 'residual
plotResiduals_age
```




```
#7
fit_dis<- lm(medv ~ dis, boston)
summary(fit_dis)
```

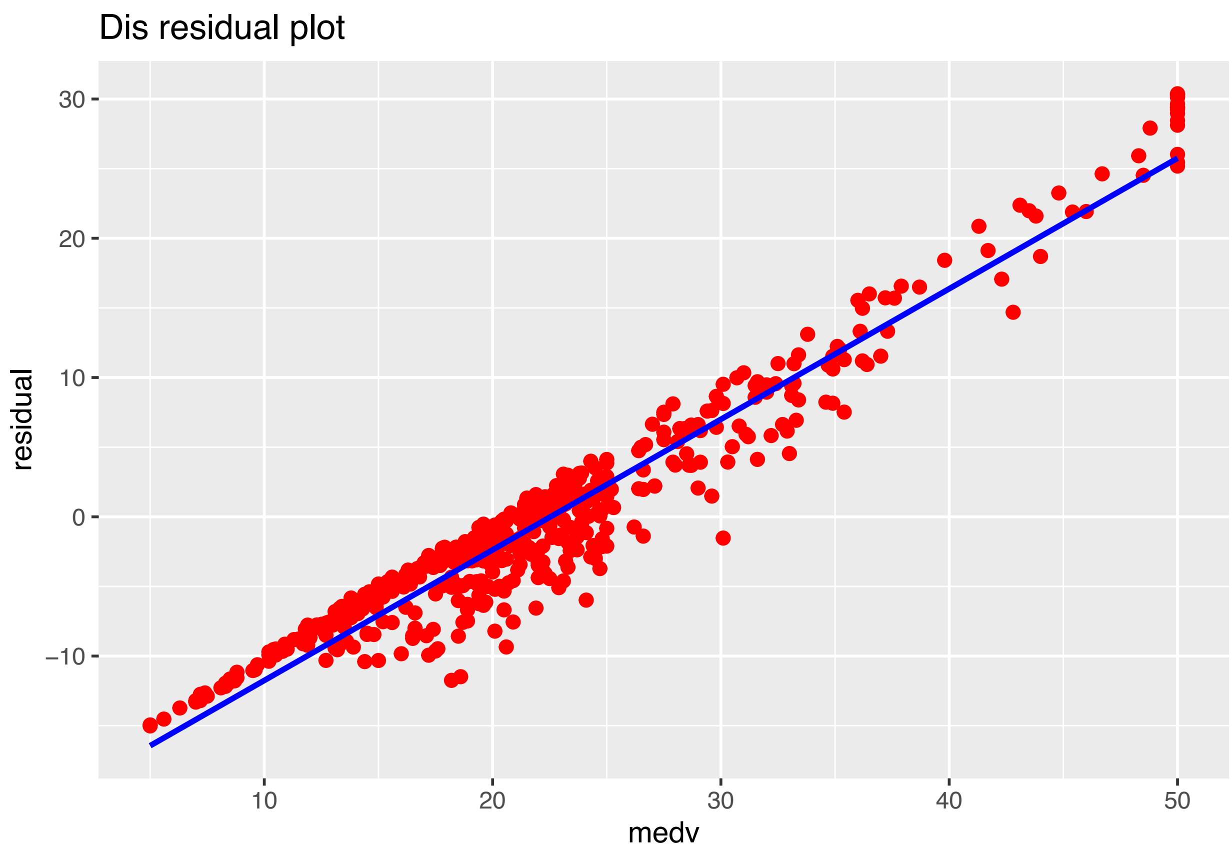
```
##
## Call:
## lm(formula = medv ~ dis, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.016  -5.556  -1.865   2.288  30.377
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.3901     0.8174  22.499  < 2e-16 ***
## dis           1.0916     0.1884   5.795 1.21e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 504 degrees of freedom
## Multiple R-squared:  0.06246,    Adjusted R-squared:  0.0606
## F-statistic: 33.58 on 1 and 504 DF,  p-value: 1.207e-08
```

```
confint(fit_dis, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_dis <- resid(fit_dis)
plotResiduals_dis <- ggplot(data = data.frame(x = boston$medv, y= residuals_dis), aes(x = x, y=y)) + ge

plotResiduals_dis <- plotResiduals_dis +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Dis residual plot", y = 'residual
plotResiduals_dis
```




```r
#8
fit_rad<- lm(medv ~ rad, boston)
summary(fit_rad)
```

```
##
## Call:
## lm(formula = medv ~ rad, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.770  -5.199  -1.967   3.321  33.292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.38213    0.56176  46.964   <2e-16 ***
## rad         -0.40310    0.04349  -9.269   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

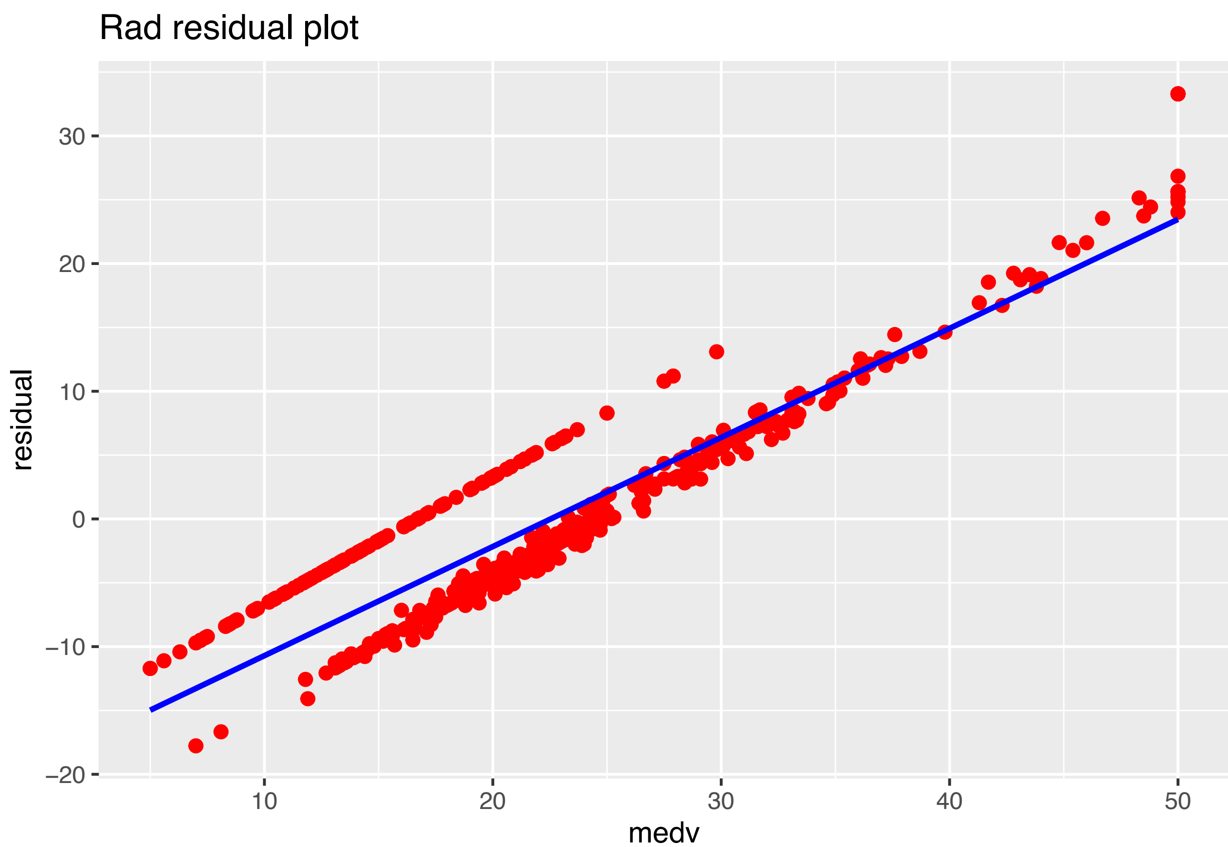
```
## Residual standard error: 8.509 on 504 degrees of freedom
## Multiple R-squared:  0.1456, Adjusted R-squared:  0.1439
## F-statistic: 85.91 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
confint(fit_rad, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```
residuals_rad <- resid(fit_rad)
plotResiduals_rad <- ggplot(data = data.frame(x = boston$medv, y= residuals_rad), aes(x = x, y=y)) + ge

plotResiduals_rad <- plotResiduals_rad +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Rad residual plot", y = 'residual
plotResiduals_rad
```




```
#9
fit_tax<- lm(medv ~ tax, boston)
summary(fit_tax)
```

```
##
## Call:
## lm(formula = medv ~ tax, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.091  -5.173  -2.085   3.158  34.058
##
```

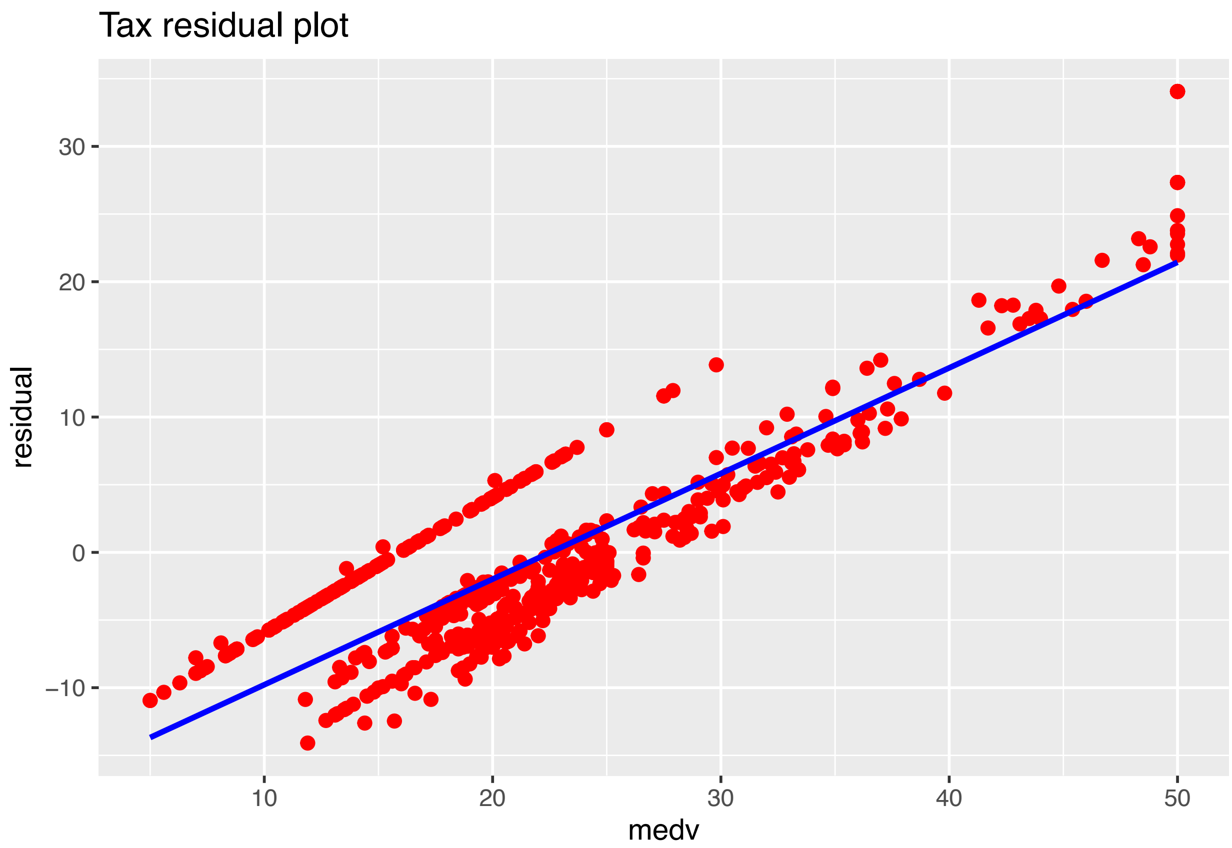
```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 32.970654   0.948296   34.77   <2e-16 ***
## tax         -0.025568   0.002147  -11.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.133 on 504 degrees of freedom
## Multiple R-squared:  0.2195, Adjusted R-squared:  0.218
## F-statistic: 141.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
confint(fit_tax, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```
residuals_tax <- resid(fit_tax)
plotResiduals_tax <- ggplot(data = data.frame(x = boston$medv, y= residuals_tax), aes(x = x, y=y)) + ge

plotResiduals_tax <- plotResiduals_tax +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Tax residual plot", y = 'residual
plotResiduals_tax
```




```
#10
fit_ptratio<- lm(medv ~ ptratio, boston)
summary(fit_ptratio)
```

```
##
```

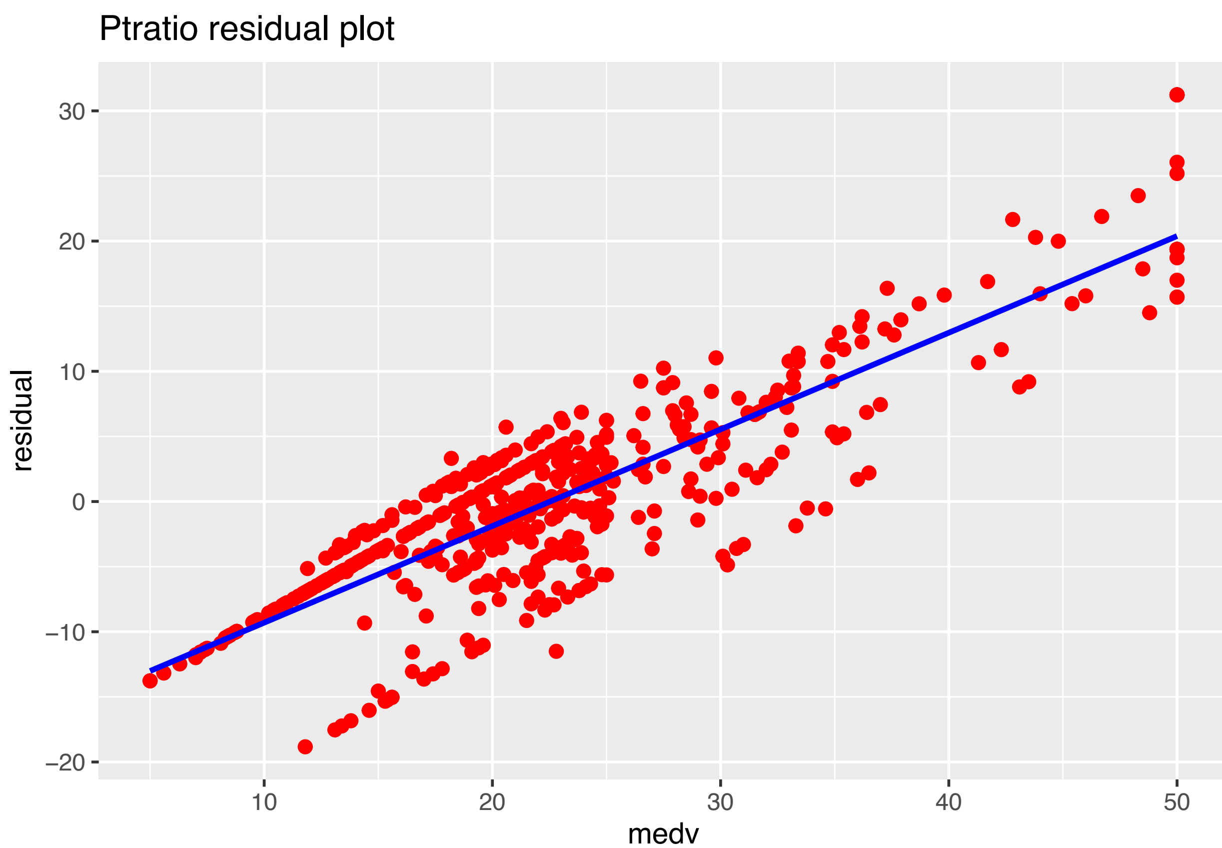
```
## Call:
## lm(formula = medv ~ ptratio, data = boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8342  -4.8262  -0.6426   3.1571  31.2303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   62.345      3.029   20.58   <2e-16 ***
## ptratio       -2.157      0.163  -13.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.931 on 504 degrees of freedom
## Multiple R-squared:  0.2578, Adjusted R-squared:  0.2564
## F-statistic: 175.1 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
confint(fit_ptratio, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_ptratio <- resid(fit_ptratio)
plotResiduals_ptratio <- ggplot(data = data.frame(x = boston$medv, y= residuals_ptratio), aes(x = x, y=y

plotResiduals_ptratio <- plotResiduals_ptratio +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Ptratio residual plot", y = 'resi
plotResiduals_ptratio
```




```r
#11
fit_black<- lm(medv ~ black, boston)
summary(fit_black)
```

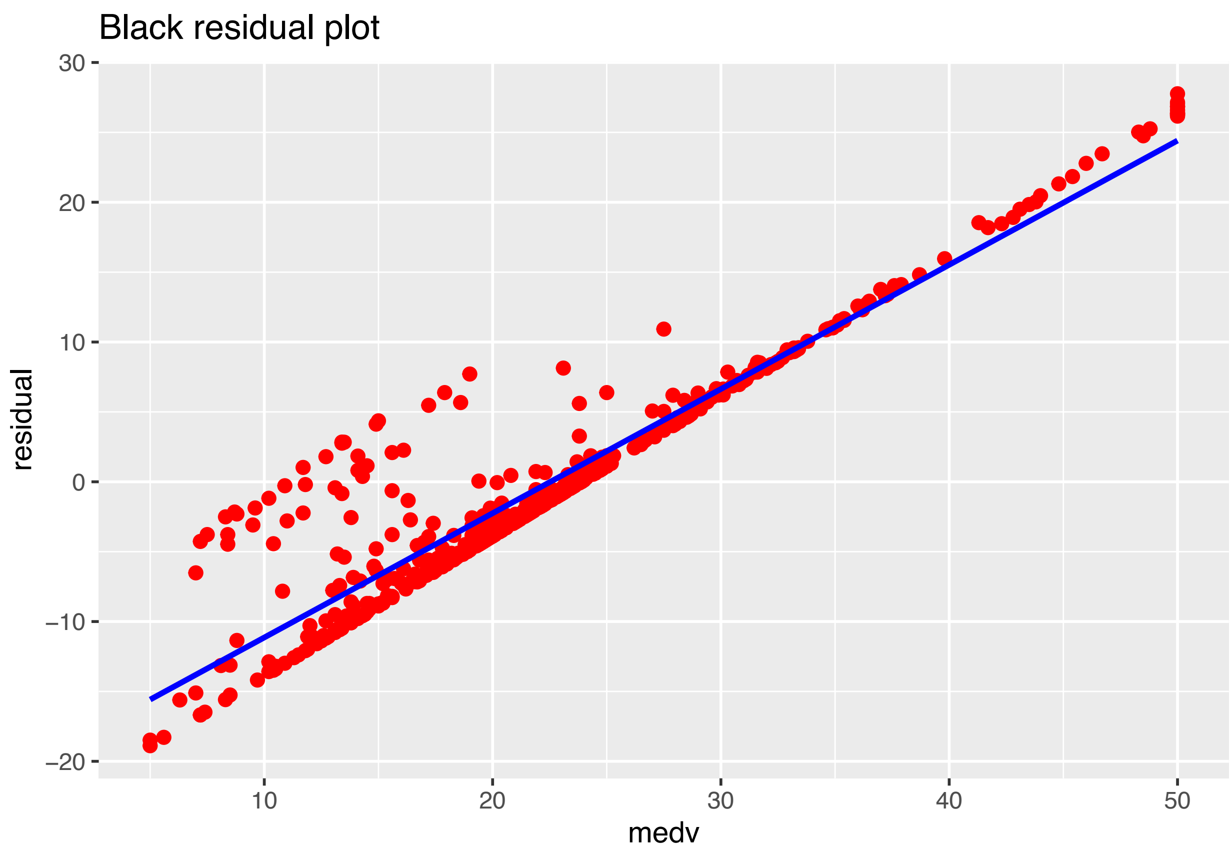
```
##
## Call:
## lm(formula = medv ~ black, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.884  -4.862  -1.684   2.932  27.763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.551034   1.557463   6.775 3.49e-11 ***
## black        0.033593   0.004231   7.941 1.32e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.679 on 504 degrees of freedom
## Multiple R-squared:  0.1112, Adjusted R-squared:  0.1094
## F-statistic: 63.05 on 1 and 504 DF,  p-value: 1.318e-14
```

```r
confint(fit_black, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_black <- resid(fit_black)
plotResiduals_black <- ggplot(data = data.frame(x = boston$medv, y= residuals_black), aes(x = x, y=y)) +

plotResiduals_black <- plotResiduals_black +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Black residual plot", y = 'residua
plotResiduals_black
```




```r
#12
fit_lstat<- lm(medv ~ lstat, boston)
summary(fit_lstat)
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

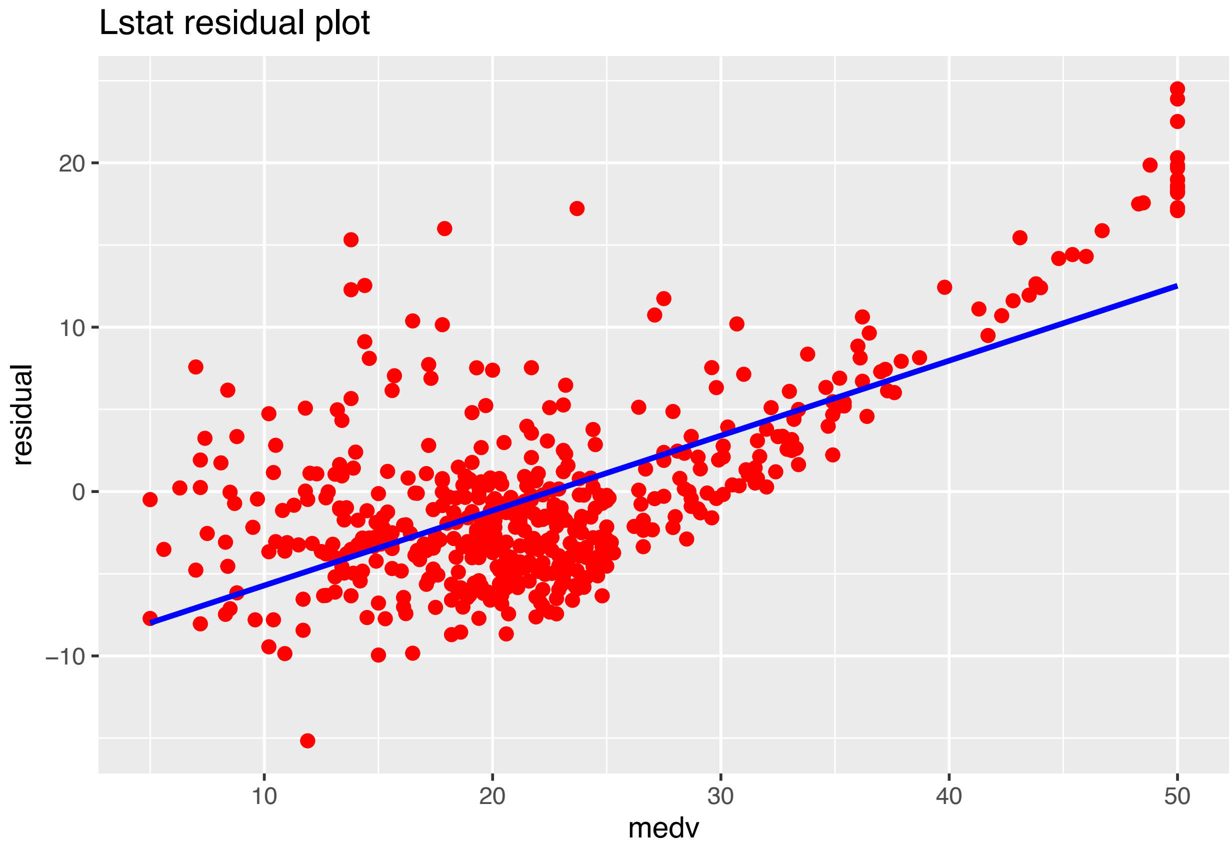
```
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
confint(fit_lstat, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```
residuals_lstat <- resid(fit_lstat)
plotResiduals_lstat <- ggplot(data = data.frame(x = boston$medv, y= residuals_lstat), aes(x = x, y=y)) +

plotResiduals_lstat <- plotResiduals_lstat +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Lstat residual plot", y = 'residua
plotResiduals_lstat
```




```
#13
fit_crim<- lm(medv ~ crim, boston)
summary(fit_crim)
```

```
##
## Call:
## lm(formula = medv ~ crim, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.957  -5.449  -2.007   2.512  29.800
##
```

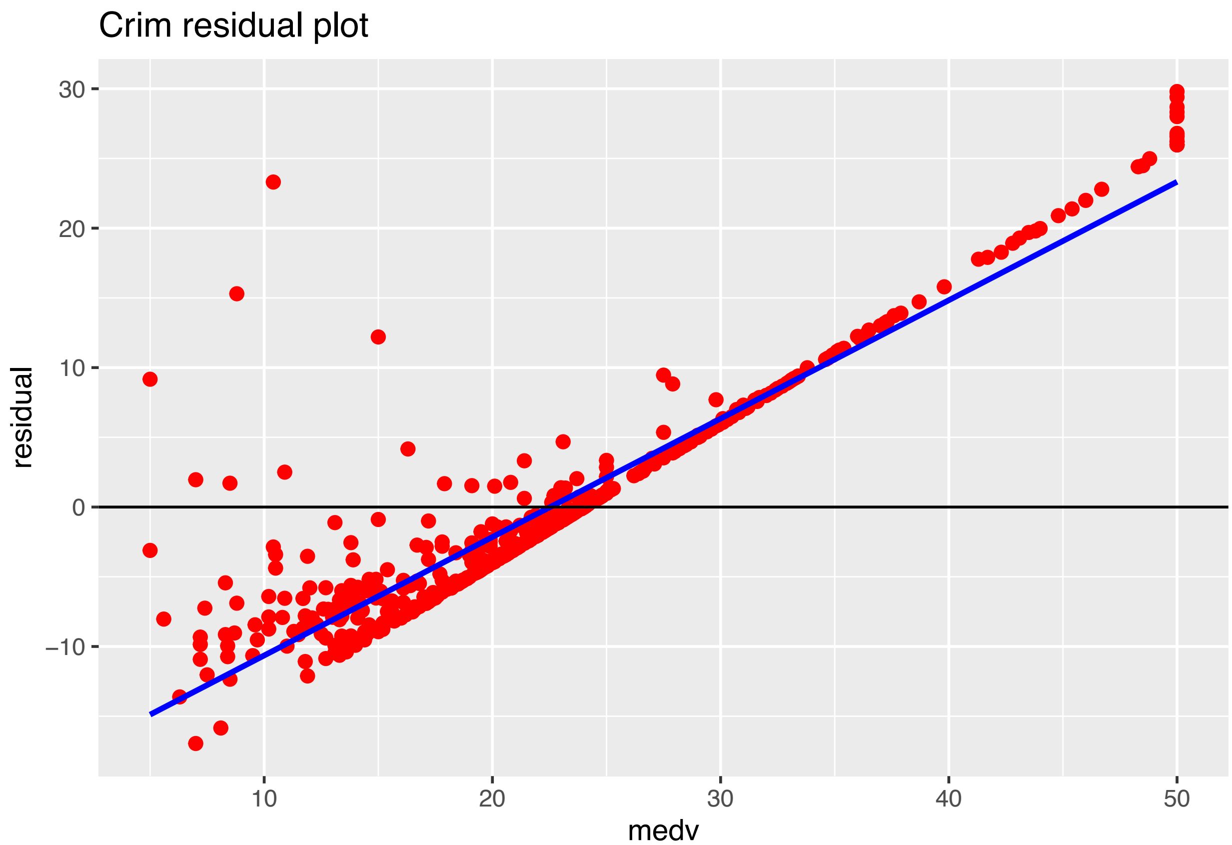
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.03311    0.40914   58.74   <2e-16 ***
## crim        -0.41519    0.04389   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.484 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
confint(fit_crim, 'indus', level = 0.95)
```

```
##       2.5 % 97.5 %
## indus    NA     NA
```

```r
residuals_crim <- resid(fit_crim)
plotResiduals_crim <- ggplot(data = data.frame(x = boston$medv, y= residuals_crim), aes(x = x, y=y)) + g

plotResiduals_crim <- plotResiduals_crim +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "Crim residual plot", y = 'residual
plotResiduals_crim
```



4. Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$?

From the table below we can see that the 'indus' and 'age' variables are not significant as their p-values(<0.05)are high and there are no '*' for those two variables.So we can reject the null hypothesis for all the other values except for the 'age' and 'indus'.

crim 0.001087 ** zn 0.000778 ***indus 0.738288***
chas 0.001925 *nox 4.25e-06* **rm < 2e-16** *age 0.958229*
dis 6.01e-13 **rad 5.07e-06** *tax 0.001112* ***ptratio 1.31e-12*** black 0.000573 ***lstat < 2e-16***

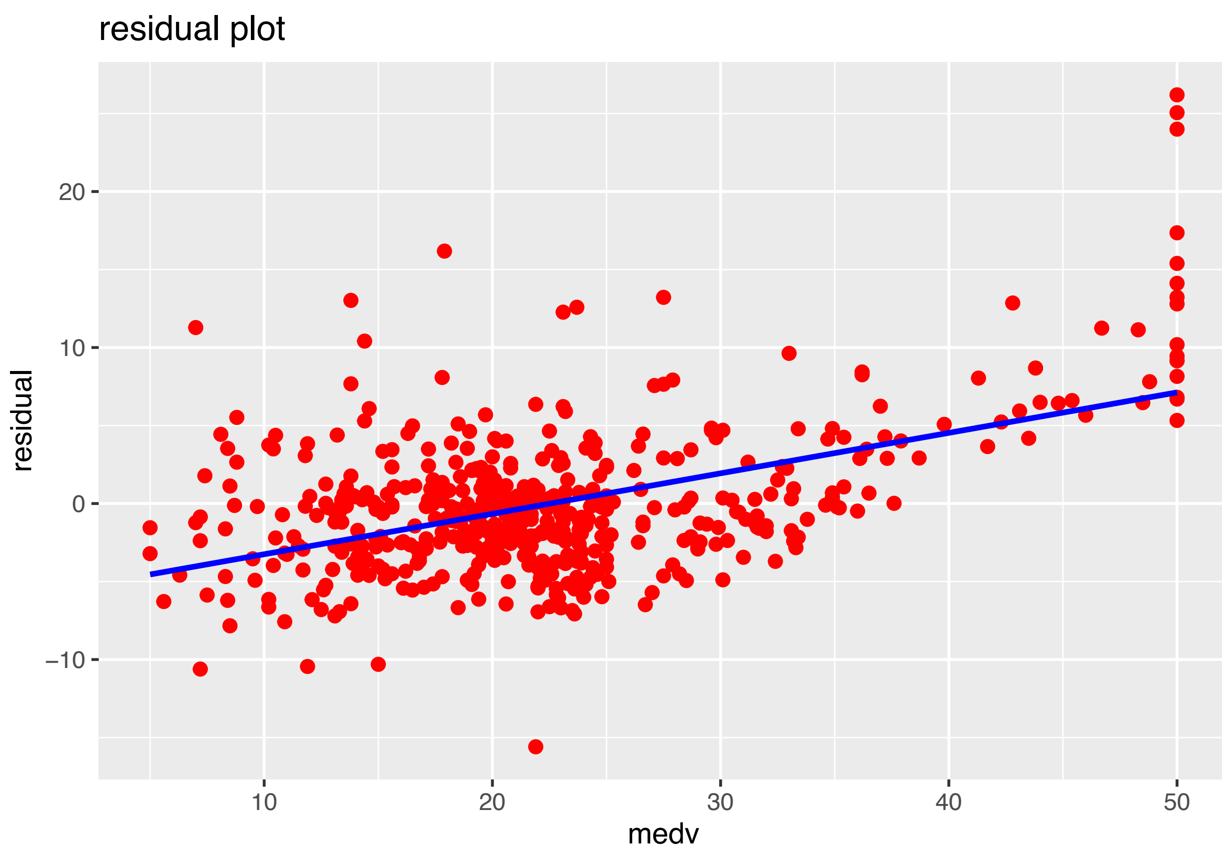
```
fit_multivariate<- lm(medv ~ crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, boston) #cr
summary(fit_multivariate)
```

```
##
## 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
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
residuals_multivariate<- resid(fit_multivariate)

plotResiduals_multivariate <- ggplot(data = data.frame(x = boston$medv, y= residuals_multivariate), aes

plotResiduals_multivariate <- plotResiduals_multivariate +
  stat_smooth(method = 'lm',se = FALSE, color = 'blue')+labs(title = "residual plot", y = 'residual', x
plotResiduals_multivariate
```



5. How do your results from (3) compare to your results from (4)? Create a plot displaying the univariate regression coefficients from (3) on the x-axis and the multiple regression coefficients from part (4) on the y-axis. Use this visualization to support your response.

SOlution: In (3), we found that all the variables came out to be significant but from (4) we observed that age and indus are not significant enough and hence with multivariate regression we got a better fit than univariate regression.

Now, we will plot the univariate vs multivariate regression coefficients. According to our observations, we have found that there is a significant difference between values of univariate regression coefficients and the values of the multivariate regression for the variables nox,rm and chas.

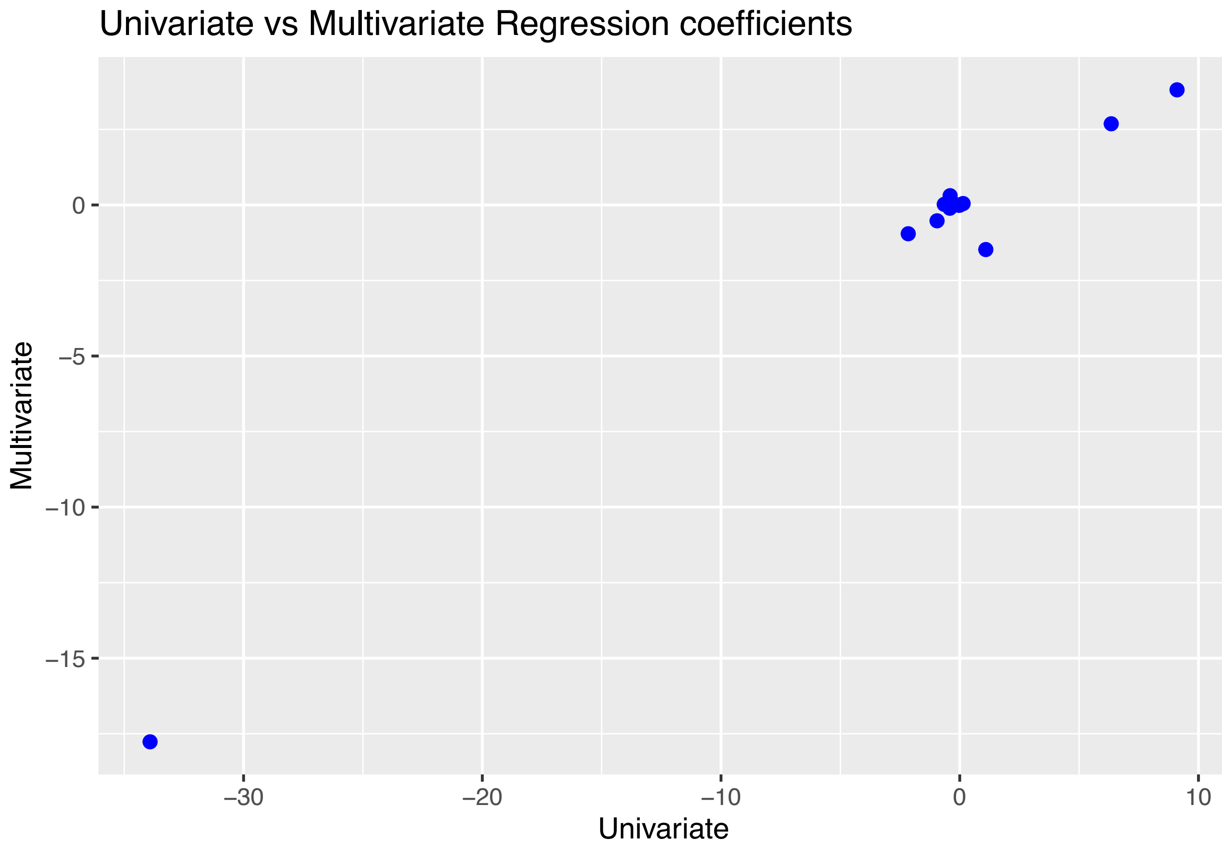
```
#creating a vector for univariate coefficient
uni <- c(coefficients(fit_crim)['crim'],coefficients(fit_zn)['zn'],coefficients(fit_indus)['indus'],coe

#creating a vector for multivariate coefficient
multi <-c(coefficients(fit_multivariate)[2:14])

#creating a dataframe for univariate and multivariate coefficients
coeff_df <-data.frame(uni,multi)

plot_uni_vs_multi <- ggplot(coeff_df, aes(uni, multi)) + geom_point(color = 'blue',size = 2)  +labs(tit

plot_uni_vs_multi
```



6. Is there evidence of a non-linear association between any of the predictors and the response? To answer this question, for each predictor X fit a model of the form:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

Solution:

Crim: For crim we have a non-linear association as the value of beta_2,beta_3 are significant. zn: For zn we have a non-linear association as the value of beta_2,beta_3 are significant. indus: For indus we have a non-linear association as the value of beta_2,beta_3 are significant. age: For age, we have no association as the value of beta_1,beta_2,beta_3 are insignificant. nox: For nox we have a non-linear association as the value of beta_3 is significant. rm: For rm we have a non-linear association as the value of beta_2,beta_3 are significant. dis: For dis we have a non-linear association as the value of beta_2,beta_3 are significant. rad: For rad we have a non-linear association as the value of beta_2,beta_3 are significant. tax: For tax, we have no association as the value of beta_1,beta_2,beta_3 are insignificant. ptratio: For ptratio, we have no association as the value of beta_1,beta_2,beta_3 are insignificant. black: For black, we have no association as the value of beta_1,beta_2,beta_3 are insignificant. lstat: For lstat we have a non-linear association as the value of beta_2,beta_3 are significant. chas: For chas we have a linear association as the value of beta_1 is significant and other are insignificant.

```
poly_crim <- lm(medv ~ poly(crim, 3, raw = TRUE), boston) #using poly function to check the beta values
summary(poly_crim)
```

```
##
## Call:
## lm(formula = medv ~ poly(crim, 3, raw = TRUE), data = boston)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.983  -4.975  -1.940   2.881  33.391
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.519e+01  4.355e-01  57.846  < 2e-16 ***
## poly(crim, 3, raw = TRUE)1 -1.136e+00  1.444e-01  -7.868 2.24e-14 ***
## poly(crim, 3, raw = TRUE)2  2.378e-02  6.808e-03   3.494 0.000518 ***
## poly(crim, 3, raw = TRUE)3 -1.489e-04  6.641e-05  -2.242 0.025411 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.159 on 502 degrees of freedom
## Multiple R-squared:  0.2177, Adjusted R-squared:  0.213
## F-statistic: 46.57 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_zn <- lm(medv ~ poly(zn, 3, raw = TRUE), boston)
summary(poly_zn)
```

```
##
## Call:
## lm(formula = medv ~ poly(zn, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.449  -5.549  -1.049   3.225  29.551
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             20.4485972  0.4359536  46.905  < 2e-16 ***
## poly(zn, 3, raw = TRUE)1  0.6433652  0.1105611   5.819 1.06e-08 ***
## poly(zn, 3, raw = TRUE)2 -0.0167646  0.0038872  -4.313 1.94e-05 ***
## poly(zn, 3, raw = TRUE)3  0.0001257  0.0000316   3.978 7.98e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.43 on 502 degrees of freedom
## Multiple R-squared:  0.1649, Adjusted R-squared:  0.1599
## F-statistic: 33.05 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_indus <- lm(medv ~ poly(indus, 3, raw = TRUE), boston)
summary(poly_indus)
```

```
##
## Call:
## lm(formula = medv ~ poly(indus, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.760  -4.725  -1.009   2.932  32.038
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                37.080160   1.663326  22.293  < 2e-16 ***
```

```
## poly(indus, 3, raw = TRUE)1 -2.806994   0.509349  -5.511 5.71e-08 ***
## poly(indus, 3, raw = TRUE)2  0.140462   0.041554   3.380 0.000781 ***
## poly(indus, 3, raw = TRUE)3 -0.002399   0.001011  -2.373 0.018026 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.844 on 502 degrees of freedom
## Multiple R-squared:  0.2768, Adjusted R-squared:  0.2725
## F-statistic: 64.06 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_nox <- lm(medv ~ poly(nox, 3, raw = TRUE), boston)
summary(poly_nox)
```

```
##
## Call:
## lm(formula = medv ~ poly(nox, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.104  -5.020  -2.144   2.747  32.416
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -22.49      38.52  -0.584   0.5596
## poly(nox, 3, raw = TRUE)1   315.10     195.10   1.615   0.1069
## poly(nox, 3, raw = TRUE)2  -615.83     320.48  -1.922   0.0552 .
## poly(nox, 3, raw = TRUE)3   350.19     170.92   2.049   0.0410 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.282 on 502 degrees of freedom
## Multiple R-squared:  0.1939, Adjusted R-squared:  0.189
## F-statistic: 40.24 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_rm <- lm(medv ~ poly(rm, 3, raw = TRUE), boston)
summary(poly_rm)
```

```
##
## Call:
## lm(formula = medv ~ poly(rm, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.102  -2.674   0.569   3.011  35.911
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              241.3108    47.3275   5.099 4.85e-07 ***
## poly(rm, 3, raw = TRUE)1 -109.3906    22.9690  -4.763 2.51e-06 ***
## poly(rm, 3, raw = TRUE)2   16.4910     3.6750   4.487 8.95e-06 ***
## poly(rm, 3, raw = TRUE)3   -0.7404     0.1935  -3.827 0.000146 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.11 on 502 degrees of freedom
```

```
## Multiple R-squared:  0.5612, Adjusted R-squared:  0.5586
## F-statistic:   214 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_age <- lm(medv ~ poly(age, 3, raw = TRUE), boston)
summary(poly_age)
```

```
##
## Call:
## lm(formula = medv ~ poly(age, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.443  -4.909  -2.234   2.185  32.944
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.893e+01  2.992e+00   9.668   <2e-16 ***
## poly(age, 3, raw = TRUE)1 -1.224e-01  2.014e-01  -0.608    0.544
## poly(age, 3, raw = TRUE)2  2.355e-03  3.930e-03   0.599    0.549
## poly(age, 3, raw = TRUE)3 -2.318e-05  2.279e-05  -1.017    0.310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.472 on 502 degrees of freedom
## Multiple R-squared:  0.1566, Adjusted R-squared:  0.1515
## F-statistic: 31.06 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_dis <- lm(medv ~ poly(dis, 3, raw = TRUE), boston)
summary(poly_dis)
```

```
##
## Call:
## lm(formula = medv ~ poly(dis, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.571  -5.242  -2.037   2.397  34.769
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.03789    2.91134   2.417  0.01599 *
## poly(dis, 3, raw = TRUE)1  8.59284    2.06633   4.158 3.77e-05 ***
## poly(dis, 3, raw = TRUE)2 -1.24953    0.41235  -3.030  0.00257 **
## poly(dis, 3, raw = TRUE)3  0.05602    0.02428   2.307  0.02146 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.727 on 502 degrees of freedom
## Multiple R-squared:  0.105,  Adjusted R-squared:  0.09968
## F-statistic: 19.64 on 3 and 502 DF,  p-value: 4.736e-12
```

```r
poly_rad <- lm(medv ~ poly(rad, 3, raw = TRUE), boston)
summary(poly_rad)
```

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

```
## lm(formula = medv ~ poly(rad, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.630  -5.151  -2.017   3.169  33.594
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 30.251303   2.567860  11.781  < 2e-16 ***
## poly(rad, 3, raw = TRUE)1 -3.799454   1.307156  -2.907 0.003815 **
## poly(rad, 3, raw = TRUE)2  0.616347   0.186057   3.313 0.000991 ***
## poly(rad, 3, raw = TRUE)3 -0.020086   0.005717  -3.514 0.000482 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.37 on 502 degrees of freedom
## Multiple R-squared:  0.1767, Adjusted R-squared:  0.1718
## F-statistic: 35.91 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_chas <- lm(medv ~ poly(chas, 3, raw = TRUE), boston)
summary(poly_chas)
```

```
##
## Call:
## lm(formula = medv ~ poly(chas, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.094  -5.894  -1.417   2.856  27.906
##
## Coefficients: (2 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   22.0938     0.4176  52.902  < 2e-16 ***
## poly(chas, 3, raw = TRUE)1    6.3462     1.5880   3.996 7.39e-05 ***
## poly(chas, 3, raw = TRUE)2        NA         NA      NA       NA
## poly(chas, 3, raw = TRUE)3        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.064 on 504 degrees of freedom
## Multiple R-squared:  0.03072,    Adjusted R-squared:  0.02879
## F-statistic: 15.97 on 1 and 504 DF,  p-value: 7.391e-05
```

```r
poly_tax <- lm(medv ~ poly(tax, 3, raw = TRUE), boston)
summary(poly_tax)
```

```
##
## Call:
## lm(formula = medv ~ poly(tax, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.109  -4.952  -1.878   2.957  33.694
##
## Coefficients:
```

```
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.222e+01  1.397e+01   3.739 0.000206 ***
## poly(tax, 3, raw = TRUE)1 -1.635e-01  1.133e-01  -1.443 0.149646
## poly(tax, 3, raw = TRUE)2  3.029e-04  2.872e-04   1.055 0.292004
## poly(tax, 3, raw = TRUE)3 -2.079e-07  2.236e-07  -0.930 0.353061
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.115 on 502 degrees of freedom
## Multiple R-squared:  0.2261, Adjusted R-squared:  0.2215
## F-statistic: 48.89 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_ptratio <- lm(medv ~ poly(ptratio, 3, raw = TRUE), boston)
summary(poly_ptratio)
```

```
##
## Call:
## lm(formula = medv ~ poly(ptratio, 3, raw = TRUE), data = boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.7795  -5.0364  -0.9778   3.4766  31.1636
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   312.28642  152.48693   2.048   0.0411 *
## poly(ptratio, 3, raw = TRUE)1 -48.69114   26.88441  -1.811   0.0707 .
## poly(ptratio, 3, raw = TRUE)2   2.83995    1.56413   1.816   0.0700 .
## poly(ptratio, 3, raw = TRUE)3  -0.05686    0.03005  -1.892   0.0590 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.898 on 502 degrees of freedom
## Multiple R-squared:  0.2669, Adjusted R-squared:  0.2625
## F-statistic: 60.91 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_black <- lm(medv ~ poly(black, 3, raw = TRUE), boston)
summary(poly_black)
```

```
##
## Call:
## lm(formula = medv ~ poly(black, 3, raw = TRUE), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.005  -4.802  -1.613   2.852  28.051
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.260e+01  2.517e+00   5.006  7.7e-07 ***
## poly(black, 3, raw = TRUE)1 -1.703e-02  6.150e-02  -0.277    0.782
## poly(black, 3, raw = TRUE)2  2.036e-04  3.258e-04   0.625    0.532
## poly(black, 3, raw = TRUE)3 -2.224e-07  4.765e-07  -0.467    0.641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 8.685 on 502 degrees of freedom
## Multiple R-squared:  0.1135, Adjusted R-squared:  0.1082
## F-statistic: 21.43 on 3 and 502 DF,  p-value: 4.463e-13
```

```
poly_lstat <- lm(medv ~ poly(lstat, 3, raw = TRUE), boston)
summary(poly_lstat)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 3, raw = TRUE), data = boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5441  -3.7122  -0.5145   2.4846  26.4153
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                48.6496253  1.4347240  33.909  < 2e-16 ***
## poly(lstat, 3, raw = TRUE)1 -3.8655928  0.3287861 -11.757  < 2e-16 ***
## poly(lstat, 3, raw = TRUE)2  0.1487385  0.0212987   6.983 9.18e-12 ***
## poly(lstat, 3, raw = TRUE)3 -0.0020039  0.0003997  -5.013 7.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.396 on 502 degrees of freedom
## Multiple R-squared:  0.6578, Adjusted R-squared:  0.6558
## F-statistic: 321.7 on 3 and 502 DF,  p-value: < 2.2e-16
```

7. Consider performing a stepwise model selection procedure to determine the best fit model. Discuss your results. How is this model different from the model in (4)?

Solution: On considering a stepwise model, we found that the backward selection is the chosen regression model and the values ‘age’ and ‘indus’ are removed from the model. First the age is removed which reduces the AIC value and then the indus is remoed which further reduces the value. After this point there is no reduction in the AIC value and hence it is the best model according to Stepwise function. We got the values which were very close to multivariate model.

```
regression_model <-lm(medv ~ ., data =boston)
step_aic_model <-stepAIC(regression_model, direction = "both")  #using stepAIC to create the stepwise r
```

```
## Start:  AIC=1589.64
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.06 11079 1587.7
## - indus    1      2.52 11081 1587.8
## <none>                 11079 1589.6
## - chas     1    218.97 11298 1597.5
## - tax      1    242.26 11321 1598.6
## - crim     1    243.22 11322 1598.6
## - zn       1    257.49 11336 1599.3
## - black    1    270.63 11349 1599.8
## - rad      1    479.15 11558 1609.1
## - nox      1    487.16 11566 1609.4
```

```
## - ptratio  1   1194.23 12273 1639.4
## - dis      1   1232.41 12311 1641.0
## - rm       1   1871.32 12950 1666.6
## - lstat    1   2410.84 13490 1687.3
##
## Step:  AIC=1587.65
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.52 11081 1585.8
## <none>                 11079 1587.7
## + age      1      0.06 11079 1589.6
## - chas     1    219.91 11299 1595.6
## - tax      1    242.24 11321 1596.6
## - crim     1    243.20 11322 1596.6
## - zn       1    260.32 11339 1597.4
## - black    1    272.26 11351 1597.9
## - rad      1    481.09 11560 1607.2
## - nox      1    520.87 11600 1608.9
## - ptratio  1   1200.23 12279 1637.7
## - dis      1   1352.26 12431 1643.9
## - rm       1   1959.55 13038 1668.0
## - lstat    1   2718.88 13798 1696.7
##
## Step:  AIC=1585.76
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 11081 1585.8
## + indus    1      2.52 11079 1587.7
## + age      1      0.06 11081 1587.8
## - chas     1    227.21 11309 1594.0
## - crim     1    245.37 11327 1594.8
## - zn       1    257.82 11339 1595.4
## - black    1    270.82 11352 1596.0
## - tax      1    273.62 11355 1596.1
## - rad      1    500.92 11582 1606.1
## - nox      1    541.91 11623 1607.9
## - ptratio  1   1206.45 12288 1636.0
## - dis      1   1448.94 12530 1645.9
## - rm       1   1963.66 13045 1666.3
## - lstat    1   2723.48 13805 1695.0
```

8. Evaluate the statistical assumptions in your regression analysis from (7) by performing a basic analysis of model residuals and any unusual observations. Discuss any concerns you have about your model.

Solution: Here, we plot the residual plot for the vairables and we found that the value of the residuals is closer to yintercept = 0 and so it is a better approach as we tend to bring the residuals closer to zero to get the best fit. There are some unusual observations which are outliers in our observations and so this is a concern related to our model. Another static assumption that we have applied here is that we used the linear regression although we have found the variables whose beta_2 and beta_3 coefficients are significants and hence we should also try to apply non-linear models to our dataset.

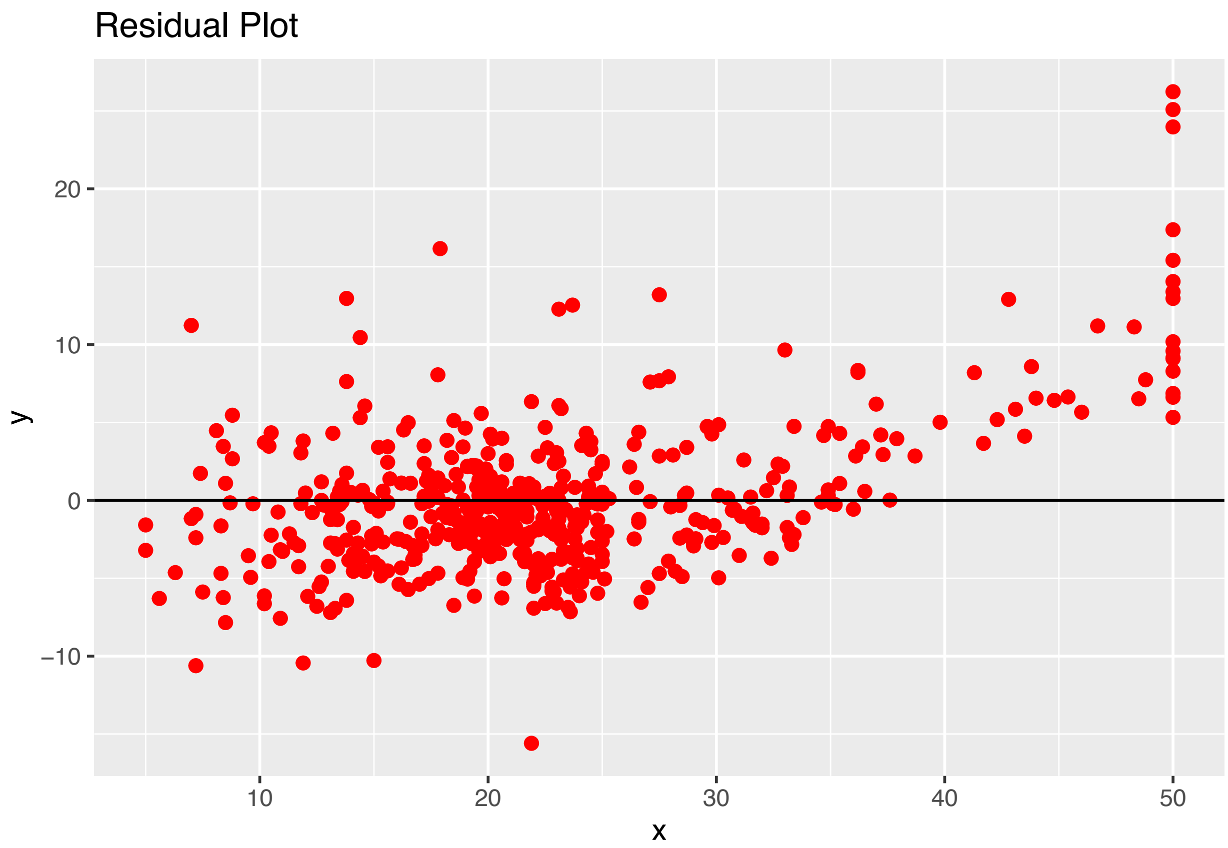
```
resid_final <-resid(stepAIC(regression_model, direction = "both")) #creating residual
```

```
## Start:  AIC=1589.64
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.06 11079 1587.7
## - indus    1      2.52 11081 1587.8
## <none>                 11079 1589.6
## - chas     1    218.97 11298 1597.5
## - tax      1    242.26 11321 1598.6
## - crim     1    243.22 11322 1598.6
## - zn       1    257.49 11336 1599.3
## - black    1    270.63 11349 1599.8
## - rad      1    479.15 11558 1609.1
## - nox      1    487.16 11566 1609.4
## - ptratio  1   1194.23 12273 1639.4
## - dis      1   1232.41 12311 1641.0
## - rm       1   1871.32 12950 1666.6
## - lstat    1   2410.84 13490 1687.3
##
## Step:  AIC=1587.65
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.52 11081 1585.8
## <none>                 11079 1587.7
## + age      1      0.06 11079 1589.6
## - chas     1    219.91 11299 1595.6
## - tax      1    242.24 11321 1596.6
## - crim     1    243.20 11322 1596.6
## - zn       1    260.32 11339 1597.4
## - black    1    272.26 11351 1597.9
## - rad      1    481.09 11560 1607.2
## - nox      1    520.87 11600 1608.9
## - ptratio  1   1200.23 12279 1637.7
## - dis      1   1352.26 12431 1643.9
## - rm       1   1959.55 13038 1668.0
## - lstat    1   2718.88 13798 1696.7
##
## Step:  AIC=1585.76
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 11081 1585.8
## + indus    1      2.52 11079 1587.7
## + age      1      0.06 11081 1587.8
## - chas     1    227.21 11309 1594.0
## - crim     1    245.37 11327 1594.8
## - zn       1    257.82 11339 1595.4
## - black    1    270.82 11352 1596.0
```

```
## - tax      1    273.62 11355 1596.1
## - rad      1    500.92 11582 1606.1
## - nox      1    541.91 11623 1607.9
## - ptratio  1   1206.45 12288 1636.0
## - dis      1   1448.94 12530 1645.9
## - rm       1   1963.66 13045 1666.3
## - lstat    1   2723.48 13805 1695.0
```

```
final_plot<-ggplot(data = data.frame(x = boston$medv, y= resid_final), aes(x = x, y=y)) + geom_point(co

final_plot
```

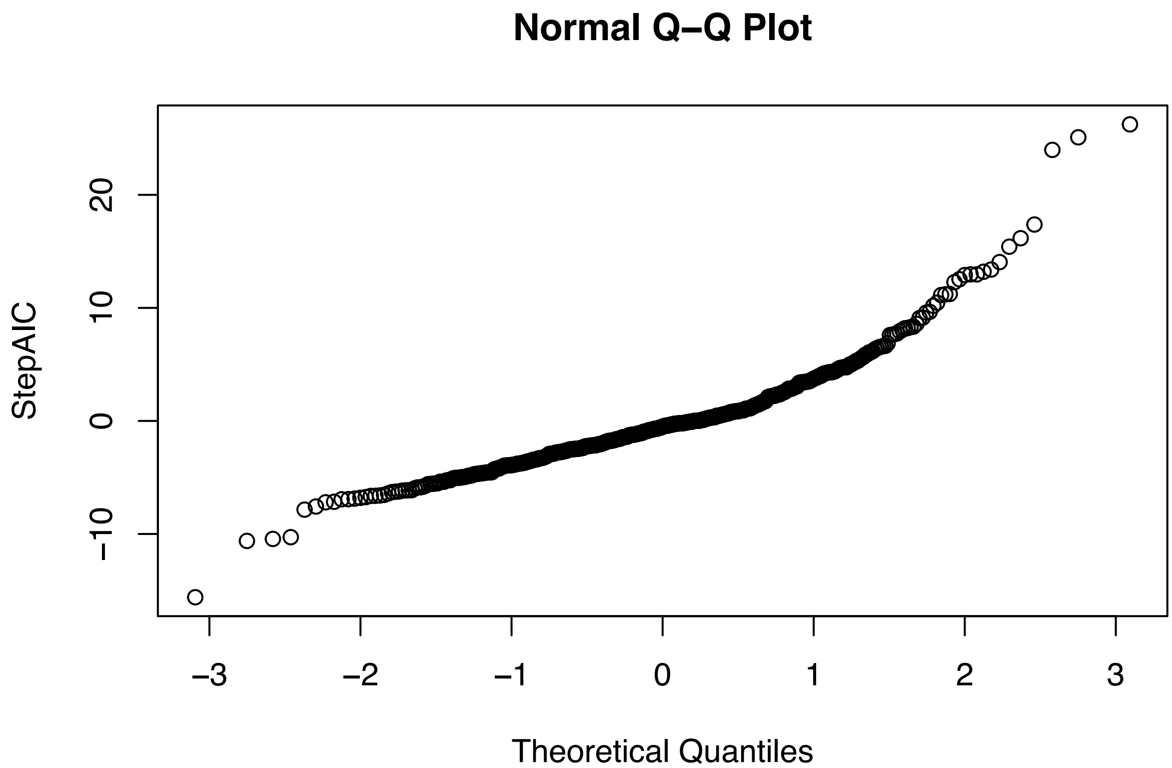



```
resid_step_wise <-  resid(stepAIC(regression_model, direction = "both"))
```

```
## Start:  AIC=1589.64
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.06 11079 1587.7
## - indus    1      2.52 11081 1587.8
## <none>                 11079 1589.6
## - chas     1    218.97 11298 1597.5
## - tax      1    242.26 11321 1598.6
## - crim     1    243.22 11322 1598.6
## - zn       1    257.49 11336 1599.3
## - black    1    270.63 11349 1599.8
```

```
## - rad      1    479.15 11558 1609.1
## - nox      1    487.16 11566 1609.4
## - ptratio  1   1194.23 12273 1639.4
## - dis      1   1232.41 12311 1641.0
## - rm       1   1871.32 12950 1666.6
## - lstat    1   2410.84 13490 1687.3
##
## Step:  AIC=1587.65
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.52 11081 1585.8
## <none>                 11079 1587.7
## + age      1      0.06 11079 1589.6
## - chas     1    219.91 11299 1595.6
## - tax      1    242.24 11321 1596.6
## - crim     1    243.20 11322 1596.6
## - zn       1    260.32 11339 1597.4
## - black    1    272.26 11351 1597.9
## - rad      1    481.09 11560 1607.2
## - nox      1    520.87 11600 1608.9
## - ptratio  1   1200.23 12279 1637.7
## - dis      1   1352.26 12431 1643.9
## - rm       1   1959.55 13038 1668.0
## - lstat    1   2718.88 13798 1696.7
##
## Step:  AIC=1585.76
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 11081 1585.8
## + indus    1      2.52 11079 1587.7
## + age      1      0.06 11081 1587.8
## - chas     1    227.21 11309 1594.0
## - crim     1    245.37 11327 1594.8
## - zn       1    257.82 11339 1595.4
## - black    1    270.82 11352 1596.0
## - tax      1    273.62 11355 1596.1
## - rad      1    500.92 11582 1606.1
## - nox      1    541.91 11623 1607.9
## - ptratio  1   1206.45 12288 1636.0
## - dis      1   1448.94 12530 1645.9
## - rm       1   1963.66 13045 1666.3
## - lstat    1   2723.48 13805 1695.0
```

```r
qqnorm(resid_step_wise, main = "Normal Q-Q Plot",
       xlab = "Theoretical Quantiles", ylab = "StepAIC",
       plot.it = TRUE, datax = FALSE)
```


Normal Q−Q Plot
StepAIC
20
10
0
−10
−3
−2
−1
0
1
2
3
Theoretical Quantiles