IMT 573: Problem Set 7 - Regression - Solutions

Malvika Mohan

Due: Tuesday, November 19, 2019

Collaborators:

Instructions:

Setup

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(tidyverse)
library(MASS) # Modern applied statistics functions
```

Housing Values in Suburbs of Boston

In this problem we will use the Boston dataset that is available in the `MASS` package. This dataset contains information about median house value for 506 neighborhoods in Boston, MA. Load this data and use it to answer the following questions.

```
data(Boston)
boston_data <- tbl_df(Boston)
head(boston_data)
```

```
## # A tibble: 6 x 14
##      crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio
##     <dbl> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <int> <dbl>   <dbl>
## 1 0.00632    18  2.31     0 0.538  6.58  65.2  4.09     1   296    15.3
## 2 0.0273      0  7.07     0 0.469  6.42  78.9  4.97     2   242    17.8
## 3 0.0273      0  7.07     0 0.469  7.18  61.1  4.97     2   242    17.8
## 4 0.0324      0  2.18     0 0.458  7.00  45.8  6.06     3   222    18.7
## 5 0.0690      0  2.18     0 0.458  7.15  54.2  6.06     3   222    18.7
## 6 0.0298      0  2.18     0 0.458  6.43  58.7  6.06     3   222    18.7
## # ... with 3 more variables: black <dbl>, lstat <dbl>, medv <dbl>
```

1. Describe the data and variables that are part of the `Boston` dataset. Tidy data as necessary. The variables are as follows : crim - Per capita crime rate by town zn - proportion of residential land zoned for lots over 25,000 square feet indus - proportion of non-retail business acres per town chas - Charles River dummy variable (value is 1 if tract bounds river) nox - Nitrogen oxide concentration per 10 million rm - average number of rooms per dwelling age - propotion of owner occupied units buits before 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 to teacher ratio by town black - the proportion of blacks by town lstat - percentage of lower status of the population medv - median value of owner-occupied homes

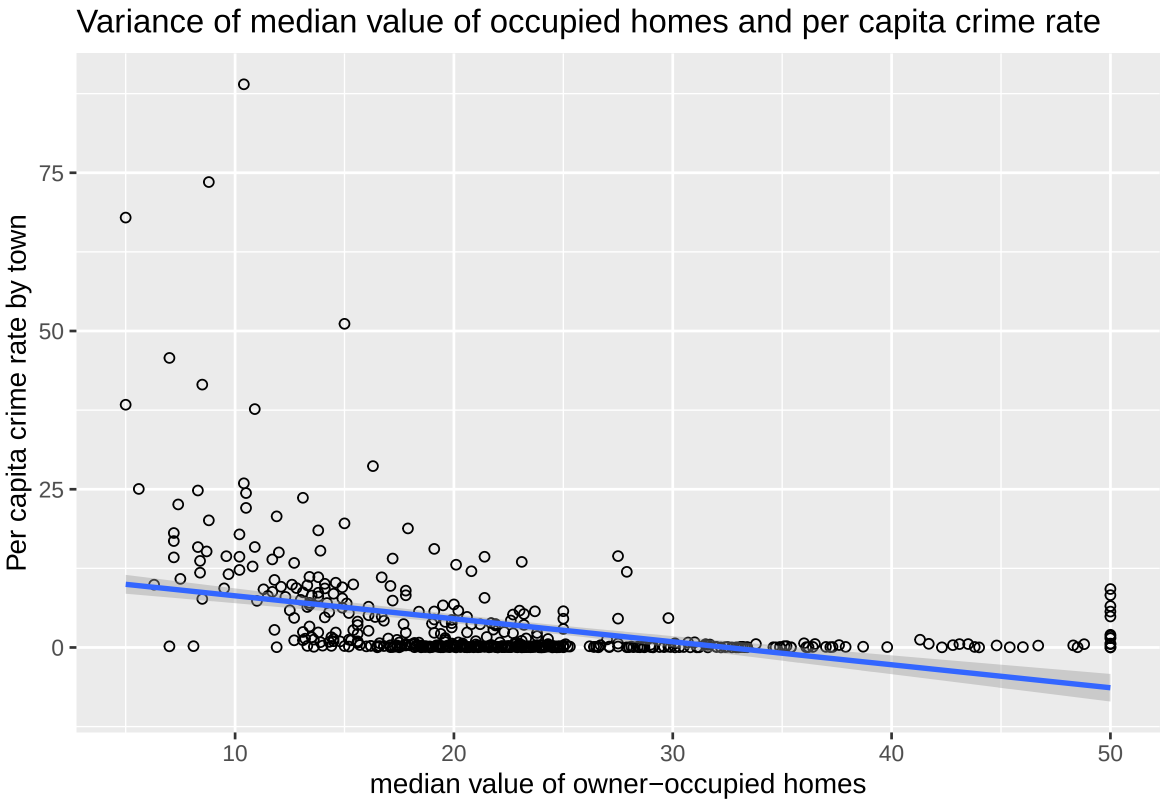
```
#Removed any NA Values present
boston_data %>%  na.omit()
```

```
## # A tibble: 506 x 14
##       crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio
##      <dbl> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <int> <dbl>   <dbl>
##  1 0.00632  18    2.31     0 0.538  6.58  65.2  4.09     1   296    15.3
##  2 0.0273    0    7.07     0 0.469  6.42  78.9  4.97     2   242    17.8
##  3 0.0273    0    7.07     0 0.469  7.18  61.1  4.97     2   242    17.8
##  4 0.0324    0    2.18     0 0.458  7.00  45.8  6.06     3   222    18.7
##  5 0.0690    0    2.18     0 0.458  7.15  54.2  6.06     3   222    18.7
##  6 0.0298    0    2.18     0 0.458  6.43  58.7  6.06     3   222    18.7
##  7 0.0883   12.5  7.87     0 0.524  6.01  66.6  5.56     5   311    15.2
##  8 0.145    12.5  7.87     0 0.524  6.17  96.1  5.95     5   311    15.2
##  9 0.211    12.5  7.87     0 0.524  5.63 100    6.08     5   311    15.2
## 10 0.170    12.5  7.87     0 0.524  6.00  85.9  6.59     5   311    15.2
## # ... with 496 more rows, and 3 more variables: black <dbl>, lstat <dbl>,
## #   medv <dbl>
```

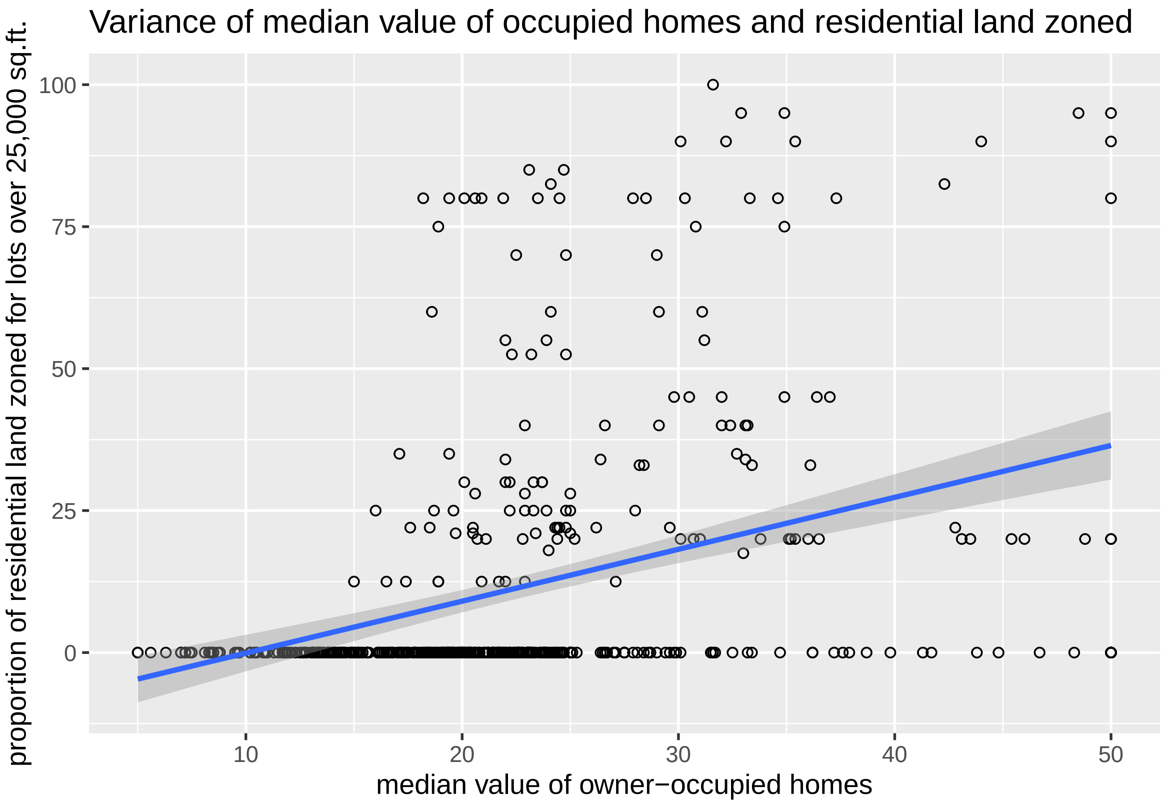
2. Consider this data in context, what is the response variable of interest? The response variable of interest is median value of owner-occuopied homes(medv).

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. There is statistically significant possitive corelation between the average number of rooms and median value of owner occupied homes since the points lie close to the regression line with only a few outliers present.While for the predictor lstat(lower status of the population) there is a significant negative corelation present.

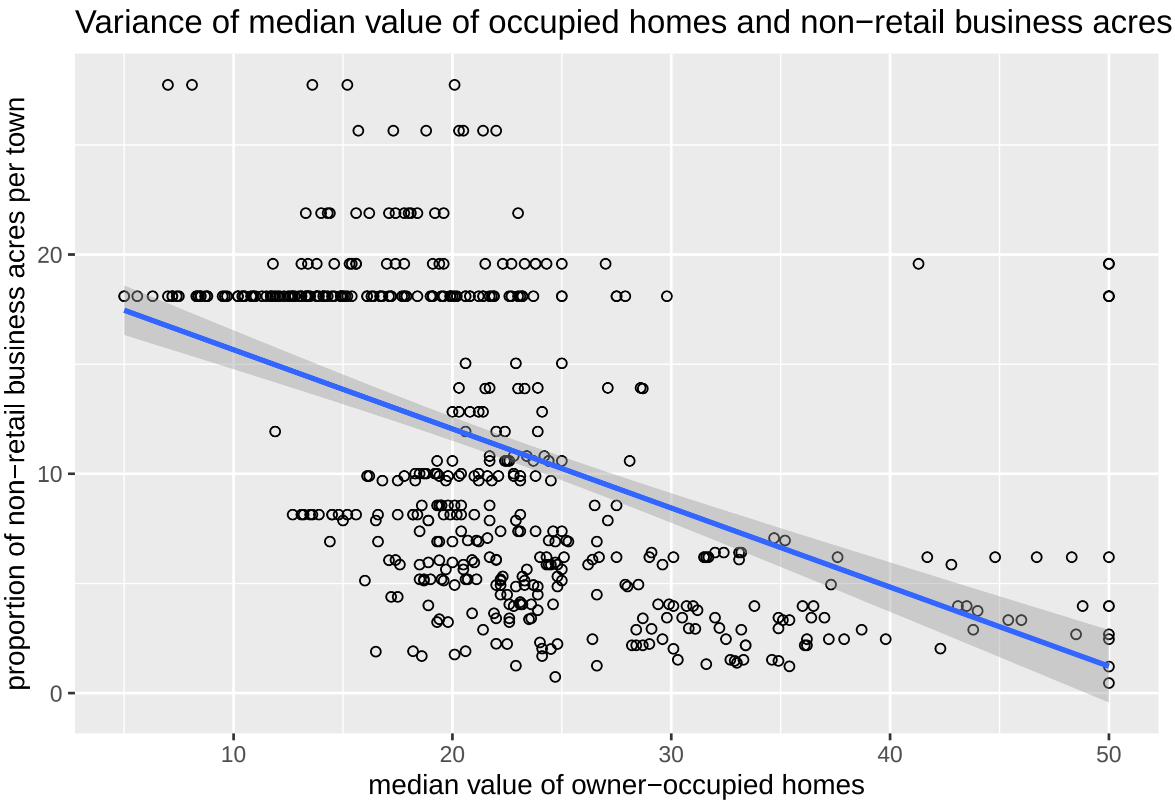
```r
#Plotting the best fit regression line for the median value of occupied homes and per capita crime
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$crim))+ geom_point(shape=1) +geom_smooth(me
```



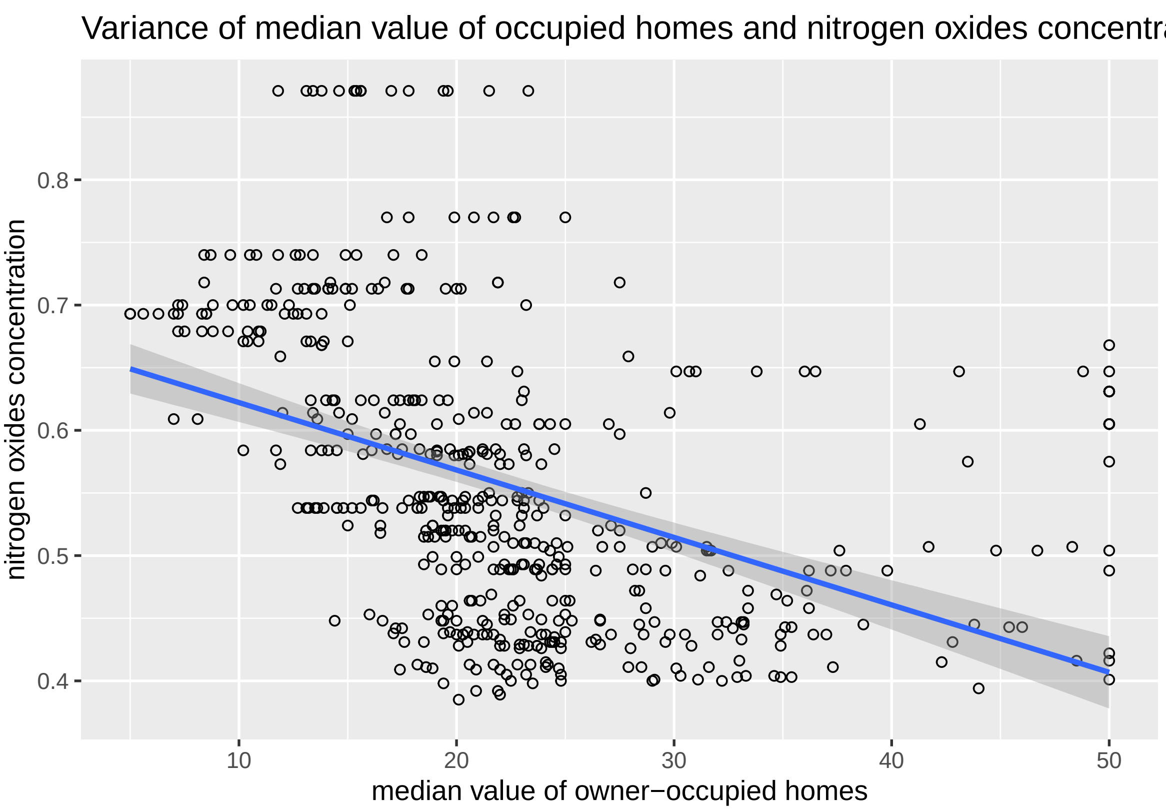
```r
#Plotting the best fit regression line for the median value of occupied homes and per capita crime
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$zn))+ geom_point(shape=1) +geom_smooth(meth
```



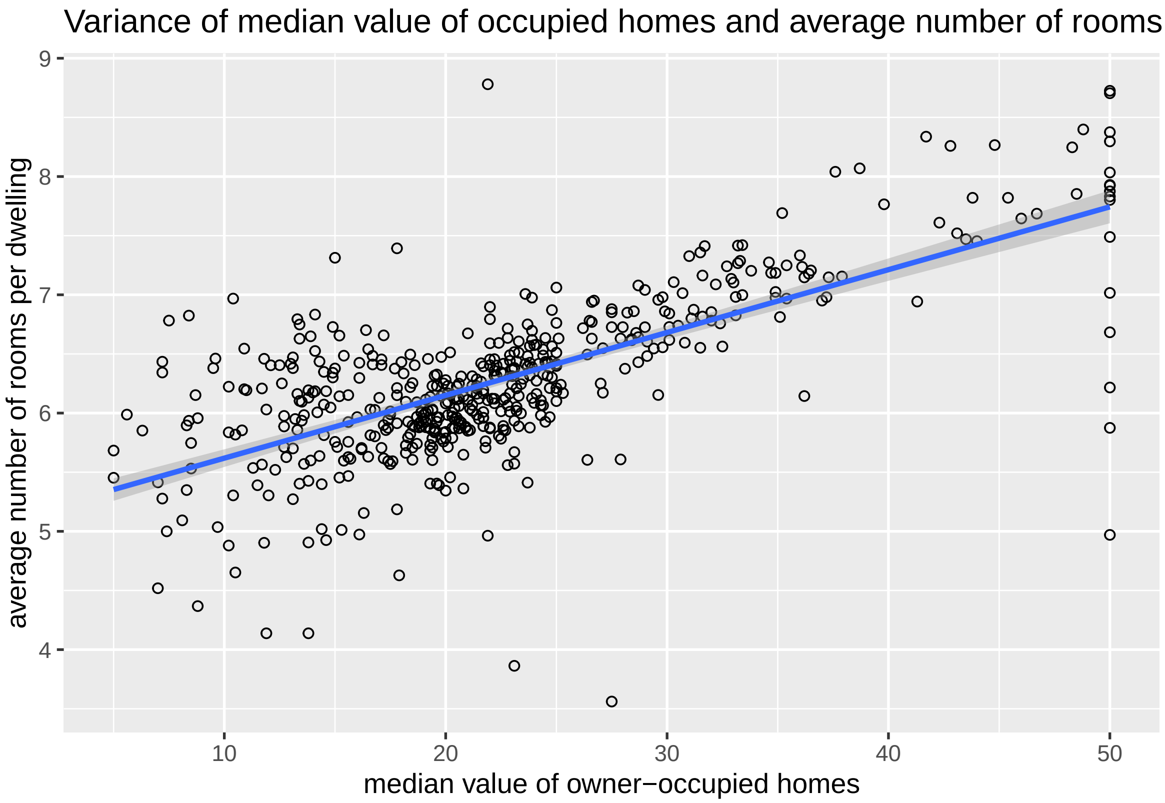

```r
#Plotting the best fit regression line for the median value of occupied homes and non-retail busine
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$indus))+ geom_point(shape=1) +geom_smooth(m
```



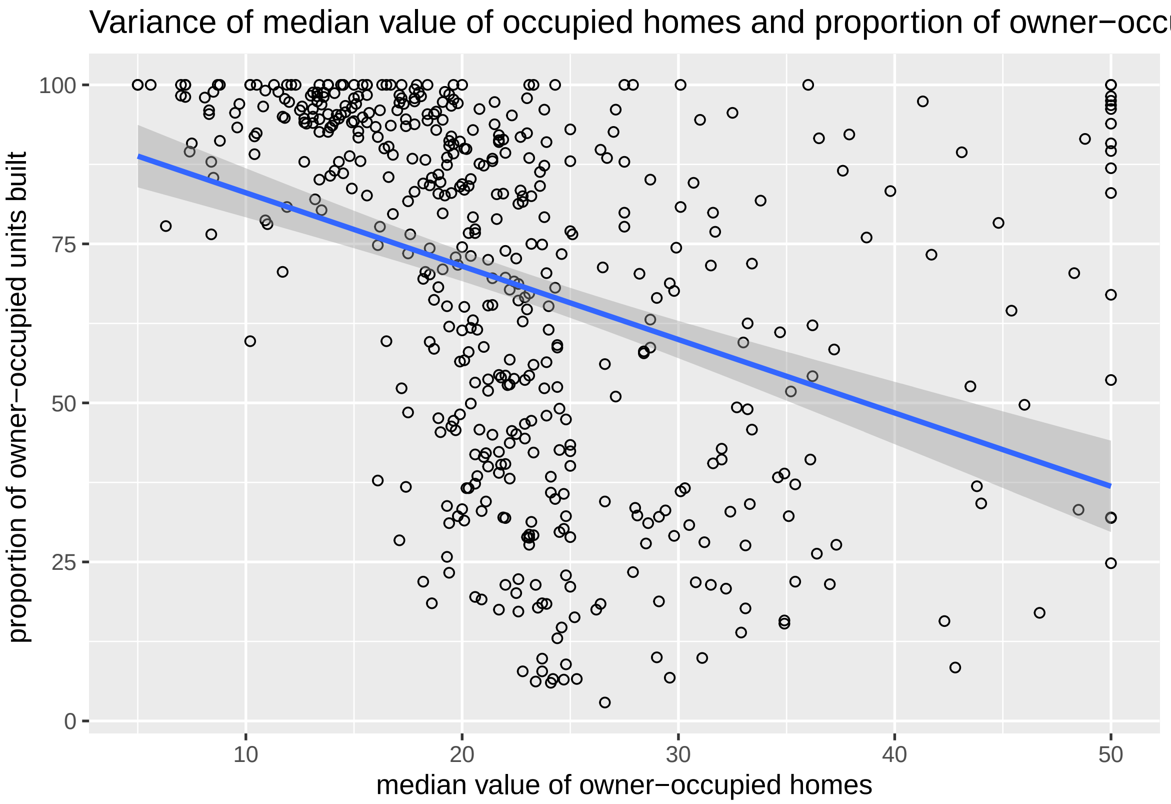

```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$nox))+ geom_point(shape=1) +geom_smooth(met
```



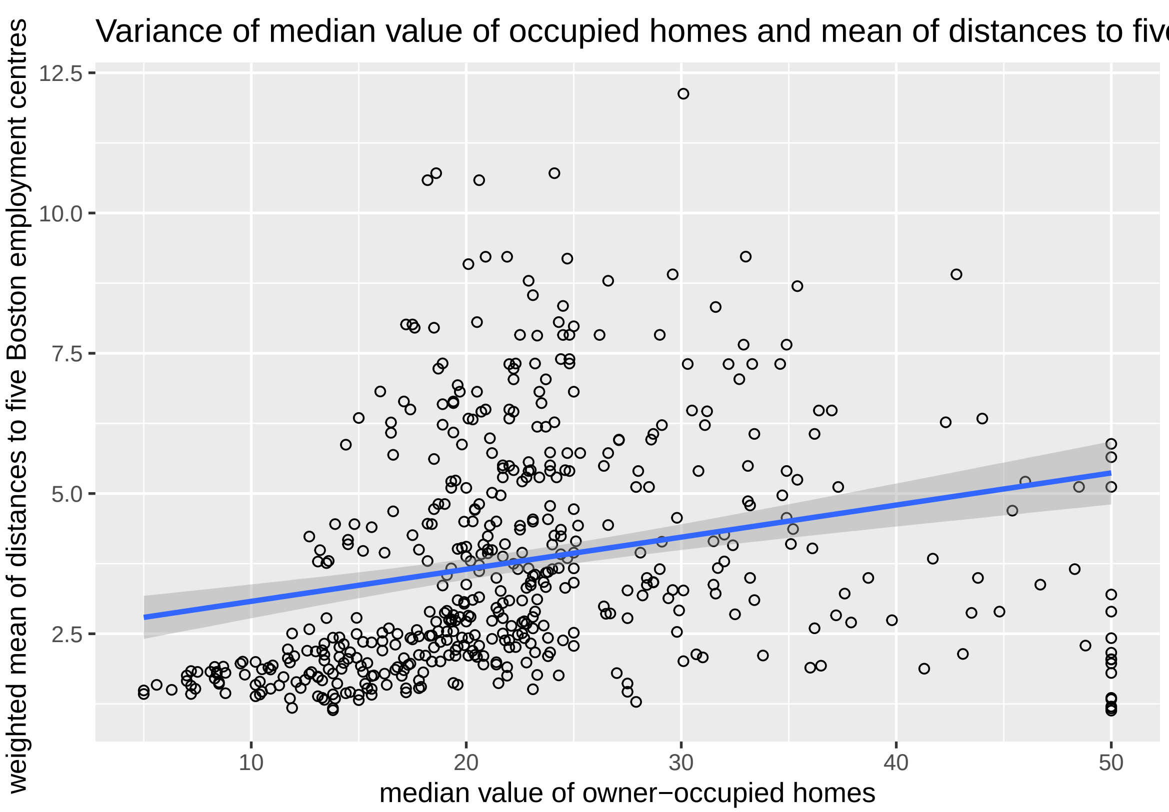
```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$rm))+ geom_point(shape=1) +geom_smooth(meth
```



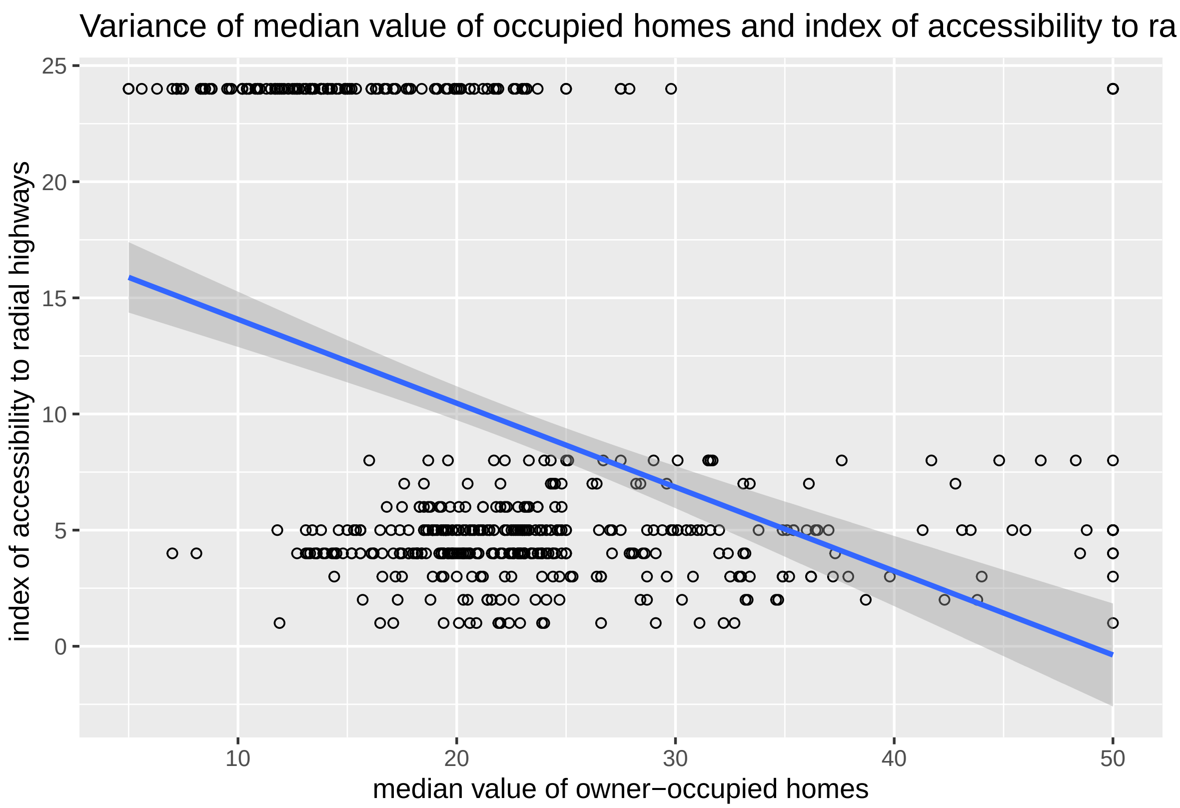

```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$age))+ geom_point(shape=1) +geom_smooth(met
```



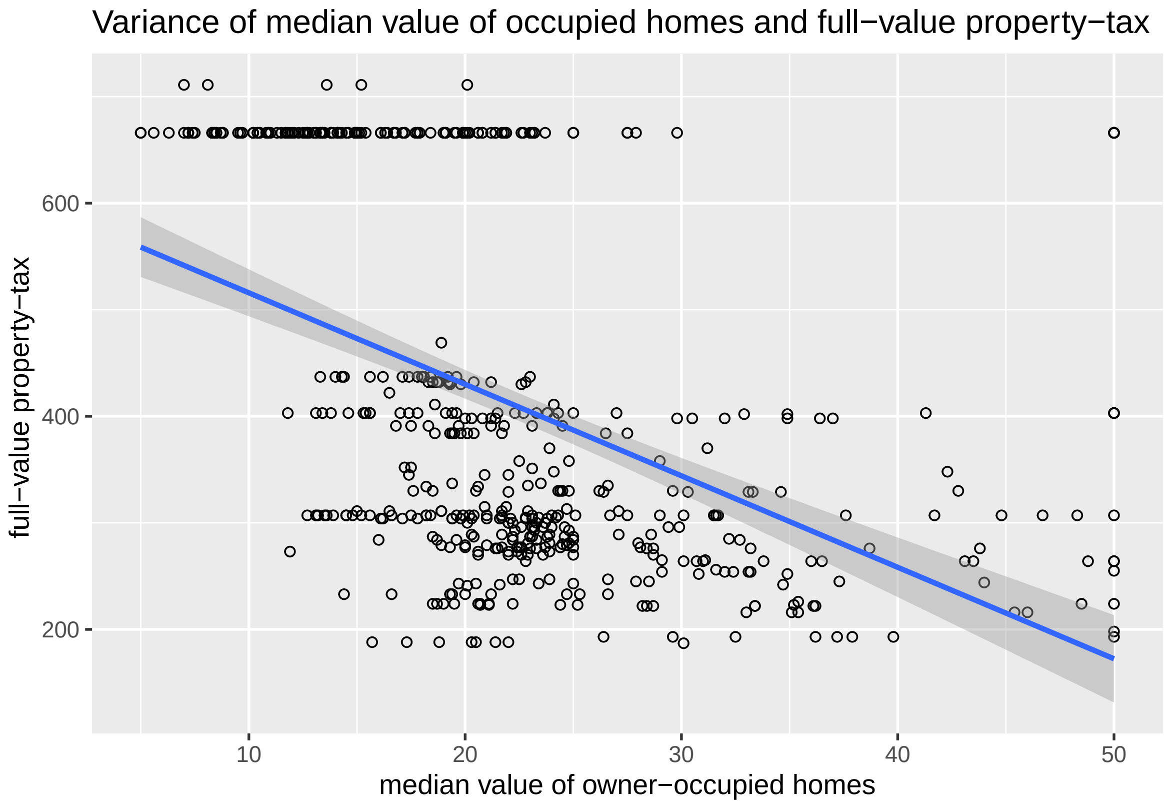

```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$dis))+ geom_point(shape=1) +geom_smooth(met
```



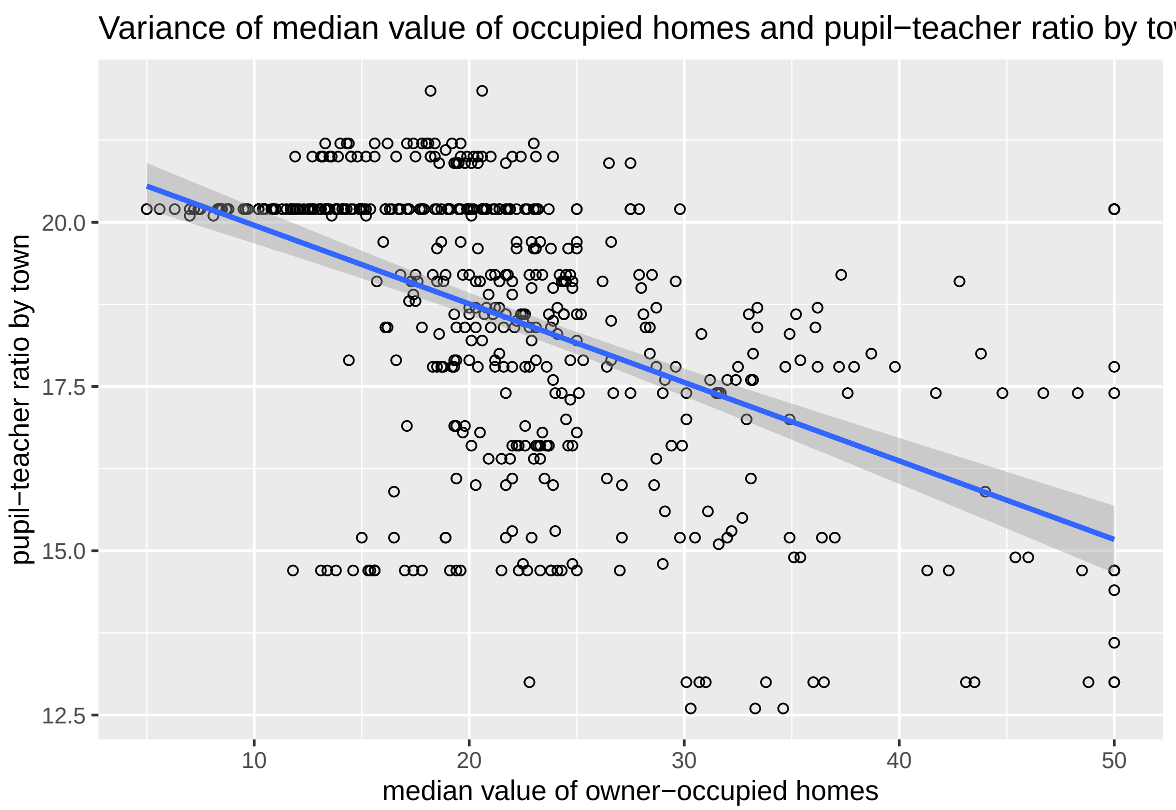
```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$rad))+ geom_point(shape=1) +geom_smooth(met
```



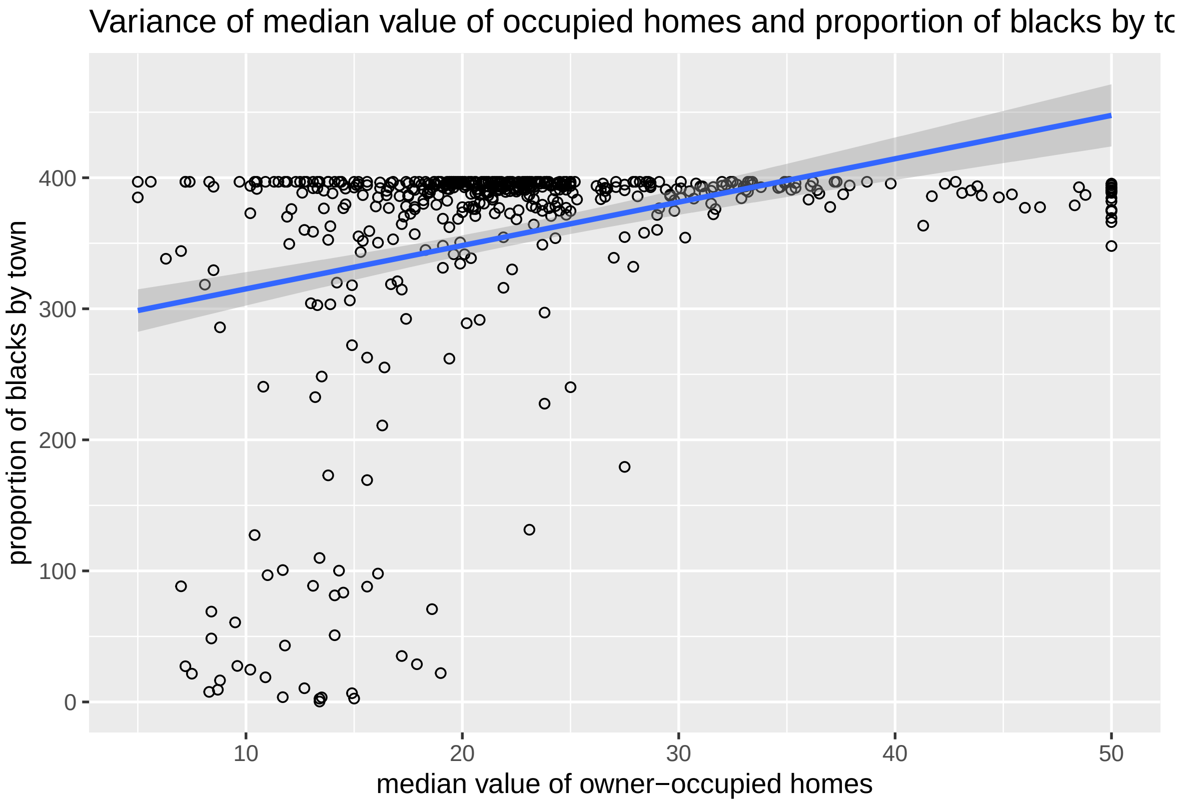

```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$tax))+ geom_point(shape=1) +geom_smooth(met
```



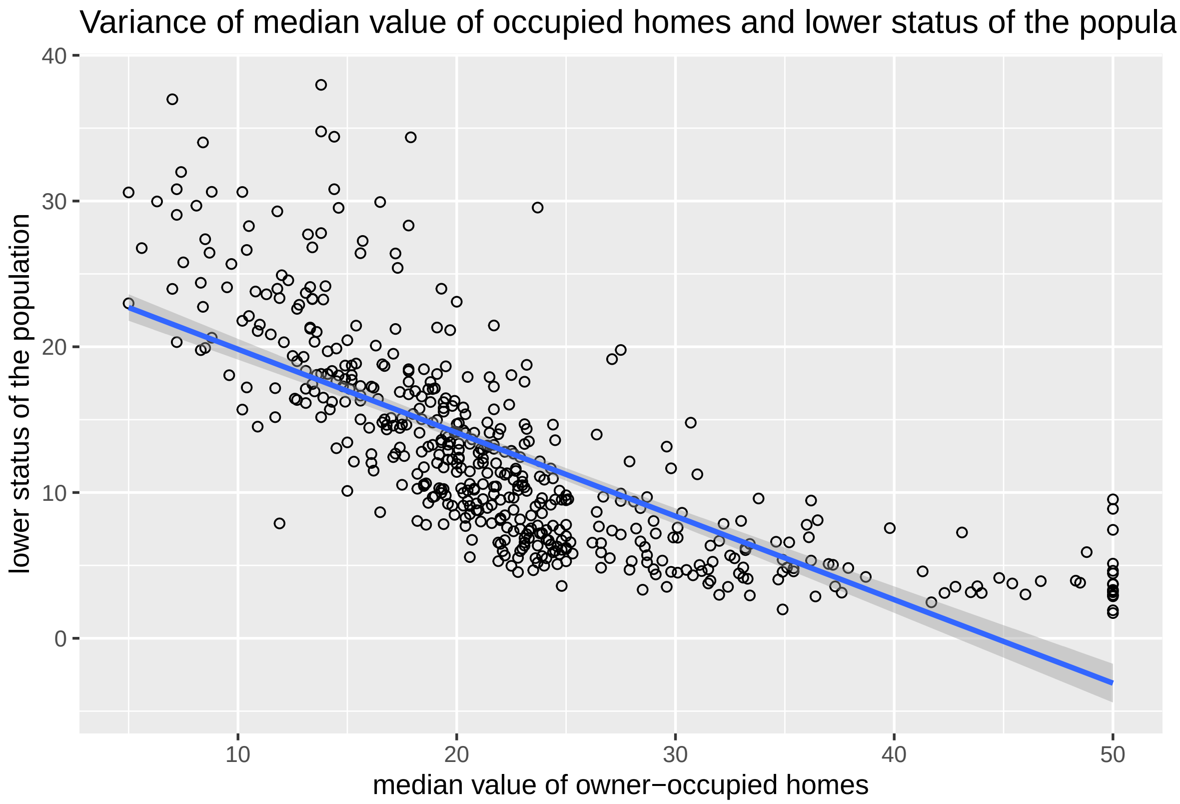

```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$ptratio))+ geom_point(shape=1) +geom_smooth
```



```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$black))+ geom_point(shape=1) +geom_smooth(m
```




```
ggplot(boston_data,aes(x=boston_data$medv,y=boston_data$lstat))+ geom_point(shape=1) +geom_smooth(m
```




```r
#Calculating co-relation between the our predictor and response variables
bivrel <- cor(boston_data,y=boston_data$medv,use = "everything",method = "pearson")
bivrel
```

```
##                [,1]
## crim    -0.3883046
## zn       0.3604453
## indus   -0.4837252
## chas     0.1752602
## nox     -0.4273208
## rm       0.6953599
## age     -0.3769546
## dis      0.2499287
## rad     -0.3816262
## tax     -0.4685359
## ptratio -0.5077867
## black    0.3334608
## lstat   -0.7376627
## medv     1.0000000
```

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 $H_0 : \beta_j = 0$? On formulating the regression model, I obtained the p value of all the predectors with respect to our response variable. For all the predictors other than age (owner-occupied units built prior to 1940) and zn (residential land zoned) we get our p-values to be less than 0.05 and hence we can reject the null hypothesis for them (crim,indus,chas,nox,rm,dis,rad,tax,pratio,black,lstat).

```r
#Storing the predictor variables in a Dummy variable
predictor_var <- subset(boston_data,select = c('crim','zn','indus','chas','nox','rm','age','dis','r

#Finding the regression model on the median value variable with all the predictors
fit_reg <- lm(boston_data$medv ~.,predictor_var)
summary(fit_reg)
```

```
##
## Call:
## lm(formula = boston_data$medv ~ ., data = predictor_var)
##
## 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
```

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. In question three, I determined the colinearity of each variable with our response variable and all the values were statistically significant. However on plotting a multivariate regression we found two values to be statistically insignificant (age and indus).

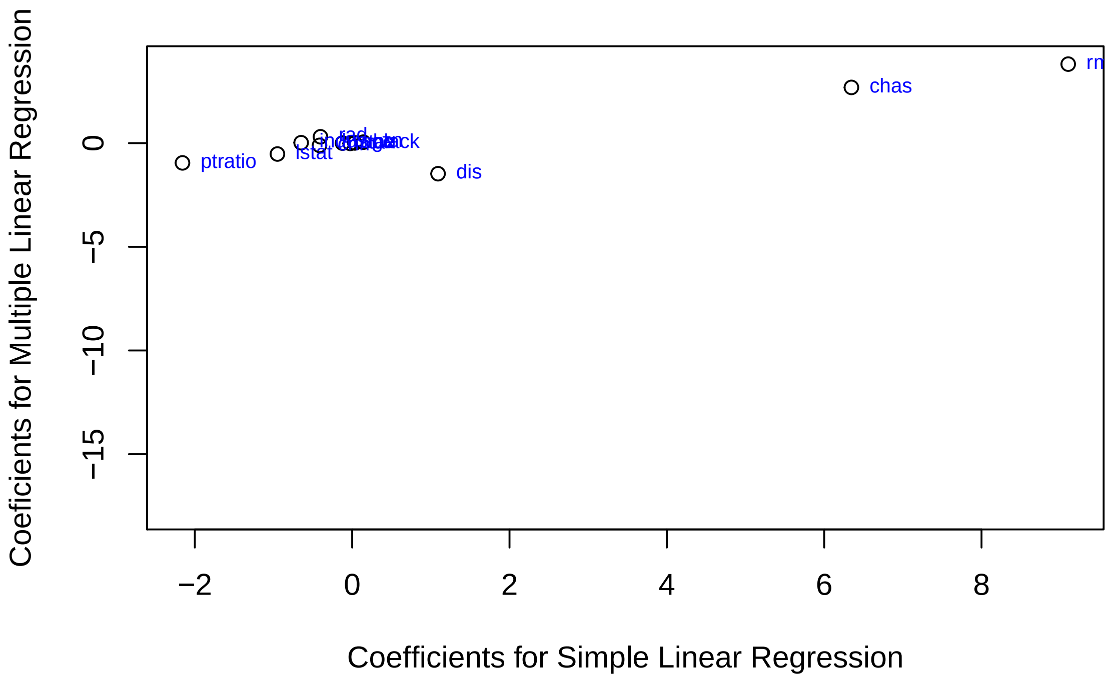
```r
#Note : I was unable to sepearate the points of the prediction variables while plotting the graph
coefs <- data.frame("predictor"=character(0), "Estimate"=numeric(0), "Std.Error"=numeric(0), "t.val
j <- 1
for(i in names(boston_data)){
  if(i != "medv"){
    #Finding the multivariable coeefficients and storing them in a coefficient matrix
    fit_reg <- summary(lm(medv ~ eval(parse(text=i)), data=boston_data))
    coefs[j,] <- c(i, fit_reg$coefficients[2,], fit_reg$r.squared)
```

```r
    j <- j+1
  }
}
#Converted all coefficients to numeric values
coefs[,-1] <- lapply(coefs[,-1], FUN=function(x) as.numeric(x))

fit_reg <- lm(boston_data$medv ~.,boston_data)
#Creating a data frame of all the
df = data.frame("multiple"=summary(fit_reg)$coefficients[-1,1])
df$simple <- NA
for(i in row.names(df)){
  #Removed the nox variable as it is displaced with respect to other points on the graph
  if(!(i %in% "nox" ))
  {
  df[row.names(df)==i, "simplecoeff"] = coefs[coefs[,1]==i, "Estimate"]
}
}
plot(df$simplecoeff , df$multiple, xlab="Coefficients for Simple Linear Regression", ylab="Coeficie
```

```
## NULL
```

```r
text(x=df$simplecoeff, y=df$multiple, labels=row.names(df), cex=.7, col="blue", pos=4)
```



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:

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

For the variables crim,zn,indus,rm,dis,rad and lstat there is evidence of non linear relationship as the squared and cubed terms of these variables is found to be statistically signficant(p value greater than 0.05). age also appears to have a non linear relationship when it is cubed as it becomes insignificant. similarly when nox is squared it appears to have a non linear relationship. For the rest of the variables there is no evidence of non linear association.

Note :On determining the polynomial coefficients for the predictor chas and our response variable we get the second and third squared values of the polynomial as NA and hence I did not use it in the polynomial regression calculation.

```
#On finding the association between the chas predictor and our response variable (as seen below) we

lm(medv ~ chas + I(chas^2) + I(chas^3), data = boston_data)
```

```
##
## Call:
## lm(formula = medv ~ chas + I(chas^2) + I(chas^3), data = boston_data)
##
## Coefficients:
## (Intercept)         chas    I(chas^2)    I(chas^3)
##      22.094        6.346           NA           NA
```

```
#Storing all predictor variables in a vector
predictor_var <- subset(boston_data,select = c('crim','zn','indus','chas','nox','rm','age','dis','r
#Creating a data frame with the coefficient details and their types
polynomial_data <- data.frame("predictor"=character(0), "Estimate"=numeric(0), "Standard Error"=num

k <- 1
#iterating over the predictor varibles
for(i in names(predictor_var)){
 if(!(i %in% c("chas"))){
#Evaluating the regression model for each of the predictors with our response variable
       print(paste0('For predictor variable : ',i))
    fit_reg <- summary(lm(medv ~ poly(eval(parse(text=i)),3), data=boston_data))
    print(fit_reg)

  }
}
```

```
## [1] "For predictor variable : crim"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                       22.5328     0.3627  62.124  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1 -80.2545     8.1589  -9.836  < 2e-16 ***
## poly(eval(parse(text = i)), 3)2  50.2416     8.1589   6.158 1.51e-09 ***
## poly(eval(parse(text = i)), 3)3 -18.2905     8.1589  -2.242   0.0254 *
```

```
## ---
## 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
##
## [1] "For predictor variable : zn"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3747  60.129  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1  74.4966     8.4296   8.837  < 2e-16 ***
## poly(eval(parse(text = i)), 3)2 -19.2591     8.4296  -2.285   0.0227 *
## poly(eval(parse(text = i)), 3)3  33.5309     8.4296   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
##
## [1] "For predictor variable : indus"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3487  64.614  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1 -99.9759     7.8445 -12.745  < 2e-16 ***
## poly(eval(parse(text = i)), 3)2  38.5184     7.8445   4.910 1.23e-06 ***
## poly(eval(parse(text = i)), 3)3 -18.6140     7.8445  -2.373    0.018 *
## ---
## 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
##
## [1] "For predictor variable : nox"
##
## Call:
```

```
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3682  61.199   <2e-16 ***
## poly(eval(parse(text = i)), 3)1 -88.3183     8.2823 -10.664   <2e-16 ***
## poly(eval(parse(text = i)), 3)2  13.8989     8.2823   1.678   0.0939 .
## poly(eval(parse(text = i)), 3)3  16.9686     8.2823   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
##
## [1] "For predictor variable : rm"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.2716  82.952  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1 143.7164     6.1103  23.520  < 2e-16 ***
## poly(eval(parse(text = i)), 3)2  52.6526     6.1103   8.617  < 2e-16 ***
## poly(eval(parse(text = i)), 3)3 -23.3832     6.1103  -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
##
## [1] "For predictor variable : age"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3766  59.830   <2e-16 ***
## poly(eval(parse(text = i)), 3)1 -77.9087     8.4717  -9.196   <2e-16 ***
```

```
## poly(eval(parse(text = i)), 3)2 -23.3290     8.4717  -2.754   0.0061 **
## poly(eval(parse(text = i)), 3)3  -8.6148     8.4717  -1.017   0.3097
## ---
## 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
##
## [1] "For predictor variable : dis"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3879  58.082  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1  51.6551     8.7267   5.919 6.00e-09 ***
## poly(eval(parse(text = i)), 3)2 -37.5859     8.7267  -4.307 1.99e-05 ***
## poly(eval(parse(text = i)), 3)3  20.1322     8.7267   2.307   0.0215 *
## ---
## 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
##
## [1] "For predictor variable : rad"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3721  60.557  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1 -78.8742     8.3700  -9.423  < 2e-16 ***
## poly(eval(parse(text = i)), 3)2 -21.4799     8.3700  -2.566 0.010568 *
## poly(eval(parse(text = i)), 3)3 -29.4095     8.3700  -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
##
## [1] "For predictor variable : tax"
```

```
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3608  62.460   <2e-16 ***
## poly(eval(parse(text = i)), 3)1 -96.8366     8.1150 -11.933   <2e-16 ***
## poly(eval(parse(text = i)), 3)2  14.9703     8.1150   1.845   0.0657 .
## poly(eval(parse(text = i)), 3)3  -7.5431     8.1150  -0.930   0.3531
## ---
## 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
##
## [1] "For predictor variable : ptratio"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                       22.5328     0.3511  64.173   <2e-16 ***
## poly(eval(parse(text = i)), 3)1 -104.9490     7.8984 -13.287   <2e-16 ***
## poly(eval(parse(text = i)), 3)2  -12.6952     7.8984  -1.607    0.109
## poly(eval(parse(text = i)), 3)3  -14.9472     7.8984  -1.892    0.059 .
## ---
## 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
##
## [1] "For predictor variable : black"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                      22.5328     0.3861  58.360  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1  68.9194     8.6851   7.935 1.38e-14 ***
## poly(eval(parse(text = i)), 3)2   9.1467     8.6851   1.053    0.293
## poly(eval(parse(text = i)), 3)3  -4.0541     8.6851  -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
##
## [1] "For predictor variable : lstat"
##
## Call:
## lm(formula = medv ~ poly(eval(parse(text = i)), 3), data = boston_data)
##
## 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)                       22.5328     0.2399  93.937  < 2e-16 ***
## poly(eval(parse(text = i)), 3)1 -152.4595     5.3958 -28.255  < 2e-16 ***
## poly(eval(parse(text = i)), 3)2   64.2272     5.3958  11.903  < 2e-16 ***
## poly(eval(parse(text = i)), 3)3  -27.0511     5.3958  -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)? The best fit model comes from taking the subset of predictors in the order as : medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn + crim + rad + tax

   We get the final AIC value as 1585.76. This model is different from model in question 4 as previously we just determined the sigificance values of our response variable (medv) with each of the predictors. However in the step wise best fit model we are finding the subset of predictors that results in a model that lowers prediction errors.It also differes from model in question 4 as we initially start with no predictors and we sequentially add the most contributive predictors and remove any variable that no longer provide an improvement in the model fit until we reach a model with the best fit.

```r
#Reference - http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/154-stepwise-
lower_reg <- lm(medv ~ 1, data = boston_data)
upper_reg <- lm(medv ~ ., data = boston_data)

#performing forward step model till all predictors are taken
step_model <- stepAIC(lower_reg , scope = list(lower = lower_reg , upper = upper_reg), direction =
```

```
## Start:  AIC=2246.51
## medv ~ 1
```

```
##
##           Df Sum of Sq   RSS    AIC
## + lstat    1   23243.9 19472 1851.0
## + rm       1   20654.4 22062 1914.2
## + ptratio  1   11014.3 31702 2097.6
## + indus    1    9995.2 32721 2113.6
## + tax      1    9377.3 33339 2123.1
## + nox      1    7800.1 34916 2146.5
## + crim     1    6440.8 36276 2165.8
## + rad      1    6221.1 36495 2168.9
## + age      1    6069.8 36647 2171.0
## + zn       1    5549.7 37167 2178.1
## + black    1    4749.9 37966 2188.9
## + dis      1    2668.2 40048 2215.9
## + chas     1    1312.1 41404 2232.7
## <none>                 42716 2246.5
##
## Step:  AIC=1851.01
## medv ~ lstat
##
##           Df Sum of Sq   RSS    AIC
## + rm       1    4033.1 15439 1735.6
## + ptratio  1    2670.1 16802 1778.4
## + chas     1     786.3 18686 1832.2
## + dis      1     772.4 18700 1832.5
## + age      1     304.3 19168 1845.0
## + tax      1     274.4 19198 1845.8
## + black    1     198.3 19274 1847.8
## + zn       1     160.3 19312 1848.8
## + crim     1     146.9 19325 1849.2
## + indus    1      98.7 19374 1850.4
## <none>                 19472 1851.0
## + rad      1      25.1 19447 1852.4
## + nox      1       4.8 19468 1852.9
## - lstat    1   23243.9 42716 2246.5
##
## Step:  AIC=1735.58
## medv ~ lstat + rm
##
##           Df Sum of Sq   RSS    AIC
## + ptratio  1    1711.3 13728 1678.1
## + chas     1     548.5 14891 1719.3
## + black    1     512.3 14927 1720.5
## + tax      1     425.2 15014 1723.5
## + dis      1     351.2 15088 1725.9
## + crim     1     311.4 15128 1727.3
## + rad      1     180.5 15259 1731.6
## + indus    1      61.1 15378 1735.6
## <none>                 15439 1735.6
## + zn       1      56.6 15383 1735.7
## + age      1      20.2 15419 1736.9
## + nox      1      14.9 15424 1737.1
## - rm       1    4033.1 19472 1851.0
## - lstat    1    6622.6 22062 1914.2
```

```
##
## Step:  AIC=1678.13
## medv ~ lstat + rm + ptratio
##
##           Df Sum of Sq   RSS    AIC
## + dis      1     499.1 13229 1661.4
## + black    1     389.7 13338 1665.6
## + chas     1     378.0 13350 1666.0
## + crim     1     122.5 13606 1675.6
## + age      1      66.2 13662 1677.7
## <none>                 13728 1678.1
## + tax      1      44.4 13684 1678.5
## + nox      1      24.8 13703 1679.2
## + zn       1      15.0 13713 1679.6
## + rad      1       6.1 13722 1679.9
## + indus    1       0.8 13727 1680.1
## - ptratio  1    1711.3 15439 1735.6
## - rm       1    3074.3 16802 1778.4
## - lstat    1    5013.6 18742 1833.7
##
## Step:  AIC=1661.39
## medv ~ lstat + rm + ptratio + dis
##
##           Df Sum of Sq   RSS    AIC
## + nox      1     759.6 12469 1633.5
## + black    1     502.6 12726 1643.8
## + chas     1     267.4 12962 1653.1
## + indus    1     242.6 12986 1654.0
## + tax      1     240.3 12989 1654.1
## + crim     1     233.5 12995 1654.4
## + zn       1     144.8 13084 1657.8
## + age      1      61.4 13168 1661.0
## <none>                 13229 1661.4
## + rad      1      22.4 13206 1662.5
## - dis      1     499.1 13728 1678.1
## - ptratio  1    1859.3 15088 1725.9
## - rm       1    2622.6 15852 1750.9
## - lstat    1    5349.2 18578 1831.2
##
## Step:  AIC=1633.47
## medv ~ lstat + rm + ptratio + dis + nox
##
##           Df Sum of Sq   RSS    AIC
## + chas     1     328.3 12141 1622.0
## + black    1     311.8 12158 1622.7
## + zn       1     151.7 12318 1629.3
## + crim     1     141.4 12328 1629.7
## + rad      1      53.5 12416 1633.3
## <none>                 12469 1633.5
## + indus    1      17.1 12452 1634.8
## + tax      1      10.5 12459 1635.0
## + age      1       0.2 12469 1635.5
## - nox      1     759.6 13229 1661.4
## - dis      1    1233.8 13703 1679.2
```

```
## - ptratio  1    2116.5 14586 1710.8
## - rm       1    2546.2 15016 1725.5
## - lstat    1    3664.3 16134 1761.8
##
## Step:  AIC=1621.97
## medv ~ lstat + rm + ptratio + dis + nox + chas
##
##           Df Sum of Sq   RSS    AIC
## + black    1     272.8 11868 1612.5
## + zn       1     164.4 11977 1617.1
## + crim     1     116.3 12025 1619.1
## + rad      1      58.6 12082 1621.5
## <none>                 12141 1622.0
## + indus    1      26.3 12115 1622.9
## + tax      1       4.2 12137 1623.8
## + age      1       2.3 12139 1623.9
## - chas     1     328.3 12469 1633.5
## - nox      1     820.4 12962 1653.1
## - dis      1    1146.8 13288 1665.6
## - ptratio  1    1924.9 14066 1694.4
## - rm       1    2480.7 14622 1714.0
## - lstat    1    3509.3 15650 1748.5
##
## Step:  AIC=1612.47
## medv ~ lstat + rm + ptratio + dis + nox + chas + black
##
##           Df Sum of Sq   RSS    AIC
## + zn       1    189.94 11678 1606.3
## + rad      1    144.32 11724 1608.3
## + crim     1     55.63 11813 1612.1
## <none>                 11868 1612.5
## + indus    1     15.58 11853 1613.8
## + age      1      9.45 11859 1614.1
## + tax      1      2.70 11866 1614.4
## - black    1    272.84 12141 1622.0
## - chas     1    289.27 12158 1622.7
## - nox      1    626.85 12495 1636.5
## - dis      1   1103.33 12972 1655.5
## - ptratio  1   1804.30 13672 1682.1
## - rm       1   2658.21 14526 1712.7
## - lstat    1   2991.55 14860 1724.2
##
## Step:  AIC=1606.31
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn
##
##           Df Sum of Sq   RSS    AIC
## + crim     1     94.71 11584 1604.2
## + rad      1     93.61 11585 1604.2
## <none>                 11678 1606.3
## + indus    1     16.05 11662 1607.6
## + tax      1      3.95 11674 1608.1
## + age      1      1.49 11677 1608.2
## - zn       1    189.94 11868 1612.5
## - black    1    298.37 11977 1617.1
```

```
## - chas     1    300.42 11979 1617.2
## - nox      1    627.62 12306 1630.8
## - dis      1   1276.45 12955 1656.8
## - ptratio  1   1364.63 13043 1660.2
## - rm       1   2384.55 14063 1698.3
## - lstat    1   3052.50 14731 1721.8
##
## Step:  AIC=1604.19
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim
##
##           Df Sum of Sq   RSS    AIC
## + rad      1    228.60 11355 1596.1
## <none>                 11584 1604.2
## + indus    1     15.77 11568 1605.5
## + age      1      2.47 11581 1606.1
## + tax      1      1.31 11582 1606.1
## - crim     1     94.71 11678 1606.3
## - black    1    222.18 11806 1611.8
## - zn       1    229.02 11813 1612.1
## - chas     1    284.34 11868 1614.5
## - nox      1    578.44 12162 1626.8
## - ptratio  1   1192.90 12776 1651.8
## - dis      1   1345.70 12929 1657.8
## - rm       1   2419.57 14003 1698.2
## - lstat    1   2753.42 14337 1710.1
##
## Step:  AIC=1596.1
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim + rad
##
##           Df Sum of Sq   RSS    AIC
## + tax      1    273.62 11081 1585.8
## <none>                 11355 1596.1
## + indus    1     33.89 11321 1596.6
## + age      1      0.10 11355 1598.1
## - zn       1    171.14 11526 1601.7
## - rad      1    228.60 11584 1604.2
## - crim     1    229.70 11585 1604.2
## - chas     1    272.67 11628 1606.1
## - black    1    295.78 11651 1607.1
## - nox      1    785.16 12140 1627.9
## - dis      1   1341.37 12696 1650.6
## - ptratio  1   1419.77 12775 1653.7
## - rm       1   2182.57 13538 1683.1
## - lstat    1   2785.28 14140 1705.1
##
## Step:  AIC=1585.76
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim + rad + tax
##
##           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. References : https://www.statisticssolutions.com/assumptions-of-multiple-linear-regression/ Firstly multiple linear regression requires the relationship between the independent and dependent variables to be linear while this may not be the case as we can see the points are spread across our regression line. Second, the multiple linear regression analysis requires that the errors between observed and predicted values should be normally distributed. Thirdly, our multiple linear regression assumes that there is no multicollinearity in the data. One concern I have with my model is that even though we see a strong significance and co-relation between some predictor variables with the response variable this may not always be true in reality.

```
residual <- resid(step_model)
plotResiduals <- ggplot(data = data.frame(x = boston_data$medv, y = residual), aes(x = x, y = y)) +
geom_point(color = 'blue', size = 1) + stat_smooth(method='lm',se=FALSE,color='red')+labs(title = '
plotResiduals
```