INFX 573: Problem Set 6 - Regression

Charudatta Deshpande

Due: Tuesday, November 21, 2017

Problem Set 6

Collaborators: Charles Hemstreet, Robert Hinshaw, Manjiri Kharkar

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Replace the "Insert Your Name Here" text in the `author:` field with your own name. List all collaborators on the top of your assignment.
2. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
3. Collaboration on problem sets is fun and useful but turn in an individual write-up in your own words and involving your own code. Do not just copy-and-paste from others' responses or code.
4. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps6.Rmd`, knit a PDF and submit the PDF file on Canvas.

1. Housing Values in Suburbs of Boston

In this problem we will use the Boston dataset that is available in *MASS* package. This dataset contains information about median house value for 506 neighborhoods in Boston, MA.

1.1 Describe data

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

Answer -

```
library(MASS)
data(Boston)
# Use 'class' command to view the type of each column in the dataset.
```

The description of Boston dataset is as follows -

This dataframe has 506 rows and 14 columns. This is primarily meant to indicate the median values of houses in Boston. Though it has some supplemental variables like crime rate, tax rate etc.

Fields -
crim - This is the per capita crime rate by town. Format - Numeric.
zn - This is the proportion of residential land zoned for lots over 25,000 sq.ft. Format - Numeric.
indus - This is the proportion of non-retail business acres per town. Format - Numeric.
chas - This is the Charles River dummy variable (= 1 if tract bounds river; 0 otherwise). Format - Integer.
nox - This is the nitrogen oxides concentration (parts per 10 million). Format - Numeric. rm - This is the

average number of rooms per dwelling. Format - Numeric.
age - This is the proportion of owner-occupied units built prior to 1940. Format - Numeric.
dis - This is the weighted mean of distances to five Boston employment centres. Format - Numeric.
rad - This is the index of accessibility to radial highways. Format - Integer.
tax - This is the full-value property-tax rate per $10,000. Format - Numeric.
ptratio - This is the pupil-teacher ratio by town. Format - Numeric.
black - This is the 1000(Bk - 0:63)2 where Bk is the proportion of blacks by town. Format - Numeric.
lstat - This is the lower status of the population (percent). Format - Numeric.
medv - This is the median value of owner-occupied homes in $1000s. Format - Numeric.

```
# Tidy the data - begin by converting it to a data.table format.
library(data.table)
as.data.table(Boston)
```

```
##         crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
##   1: 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
##   2: 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
##   3: 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
##   4: 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
##   5: 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
##  ---
## 502: 0.06263  0 11.93    0 0.573 6.593 69.1 2.4786   1 273    21.0 391.99
## 503: 0.04527  0 11.93    0 0.573 6.120 76.7 2.2875   1 273    21.0 396.90
## 504: 0.06076  0 11.93    0 0.573 6.976 91.0 2.1675   1 273    21.0 396.90
## 505: 0.10959  0 11.93    0 0.573 6.794 89.3 2.3889   1 273    21.0 393.45
## 506: 0.04741  0 11.93    0 0.573 6.030 80.8 2.5050   1 273    21.0 396.90
##      lstat medv
##   1:  4.98 24.0
##   2:  9.14 21.6
##   3:  4.03 34.7
##   4:  2.94 33.4
##   5:  5.33 36.2
##  ---
## 502:  9.67 22.4
## 503:  9.08 20.6
## 504:  5.64 23.9
## 505:  6.48 22.0
## 506:  7.88 11.9
```

```
#
#After visual inspection and use of 'summary' function, no other tidying
#of the data is necessary. For the purpose of plots or regressions, data will
#be reorganized as needed.
```

1.2 Variable of interest

Consider this data in context, what is the response variable of interest? Discuss how you think some of the possible predictor variables might be associated with this response.

Answer -

The variable of interest is 'medv' which the median value of house. Without running any statistical analyses, we can discuss following possible predictor variables.

crim - Crime - House prices are expected to be higher in areas of lower crime.

rm - Number of rooms - House prices are expected to be higher if it has more number of rooms. Without looking at the area of the house, or room sizes, establishing a relationship can be challenging.

age - Age of the house - House prices are expected to be higher if they are newer.

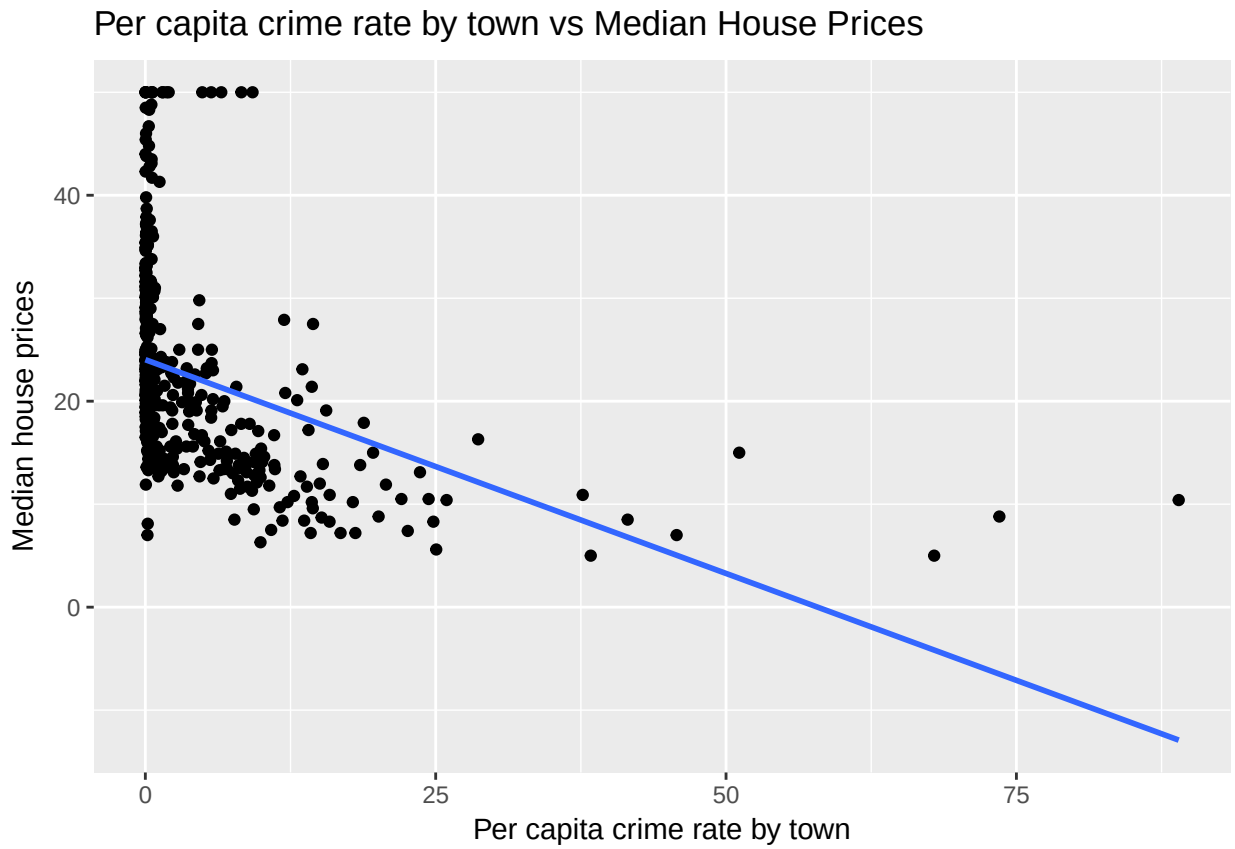
dis - weighted mean of distances to five Boston employment centres - House prices are expected to be higher if they are closer to the employment centers.

black - Percentage of black population - Ideally this factor should not affect house prices but we will look to see if a relationship exists.

1.3 Simple Regression

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.

```
# Run linear regressions with above five predictor variables and variable
#of interest.
library(ggplot2)
#
# 1. Per capita crime rate by town vs Median House Prices
#
ggplot(Boston, aes(crim, medv)) +
labs(x="Per capita crime rate by town", y="Median house prices",
     title = "Per capita crime rate by town vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```




```
m1 <- lm(medv ~ crim, data=Boston)
summary(m1)
```

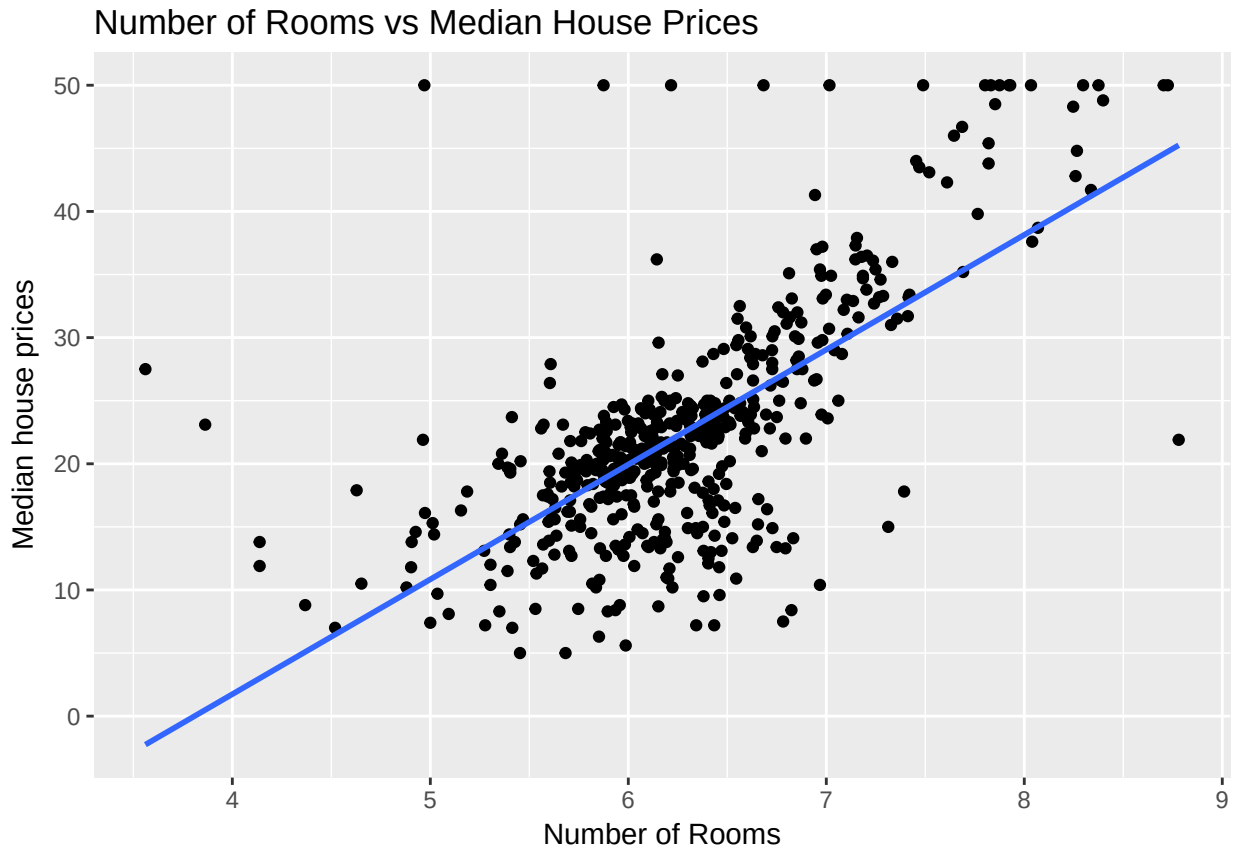
```
##
## Call:
## lm(formula = medv ~ crim, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.957  -5.449  -2.007   2.512  29.800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.03311    0.40914   58.74   <2e-16 ***
## crim        -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
```

Answer - Per capita crime rate by town vs Median House Prices

The p-value is 2.2e-16, which indicates a statistically significant association. However, the Multiple R-squared value is 0.1508 which indicates that our model does a poor job at explaining the response variable. F-statistic value is 89.49 which is far greater than 1, which also indicates a statistically significant association.

Thus we reject the null hypothesis and conclude that relationship between Per capita crime rate by town and Median House Prices is statistically significant, and we note the need for a better model.

```
#
# 2. Number of rooms vs Median House Prices
#
ggplot(Boston, aes(rm, medv)) +
labs(x="Number of Rooms", y="Median house prices",
     title = "Number of Rooms vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```




```
m2 <- lm(medv ~ rm, data=Boston)
summary(m2)
```

```
##
## 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 ***
```

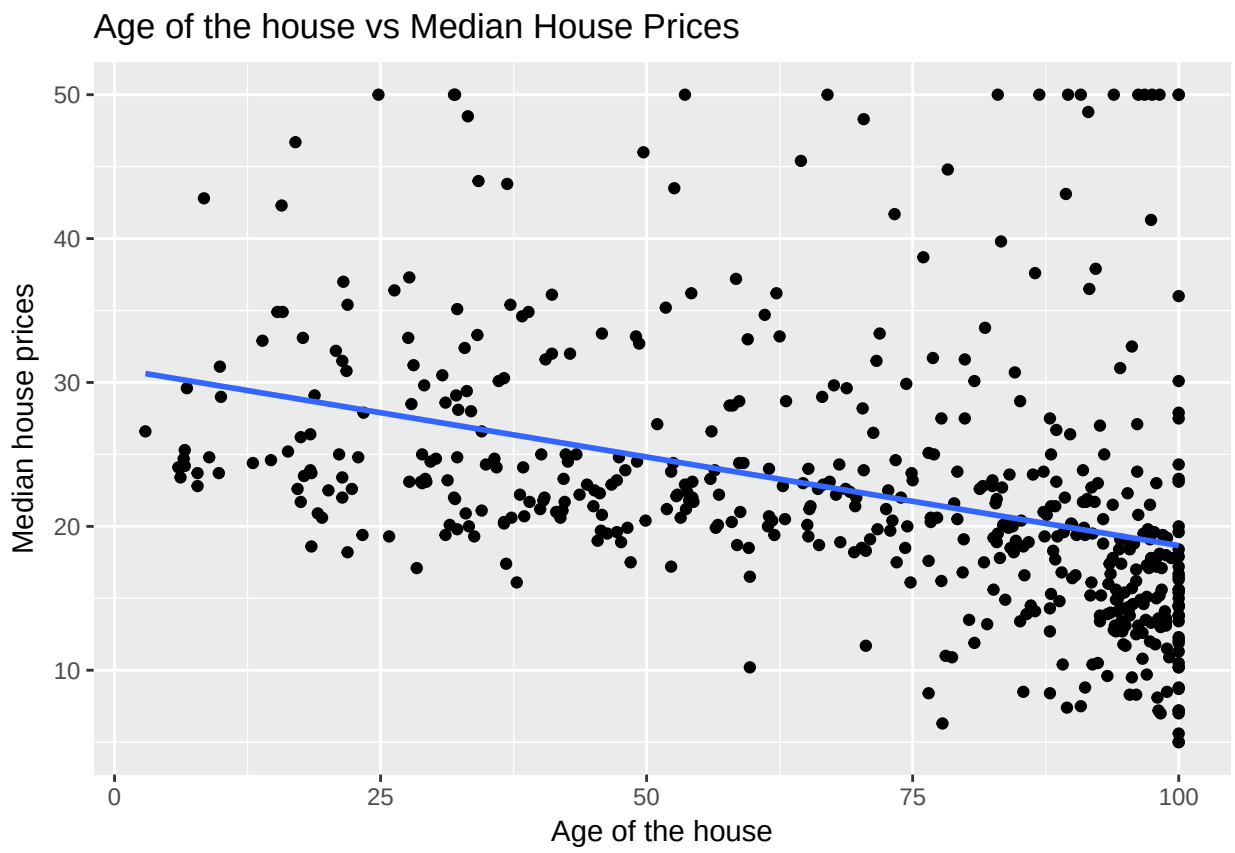
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.616 on 504 degrees of freedom
## Multiple R-squared:  0.4835, Adjusted R-squared:  0.4825
## F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

Answer - Number of rooms vs Median House Prices

The p-value is 2e-16, which indicates a statistically significant association. The Multiple R-squared value is 0.4835 which indicates that our model does a decent job at explaining the response variable. F-statistic value is 471.8 which is far greater than 1, which also indicates a statistically significant association.

Thus we reject the null hypothesis and conclude that relationship between Number of rooms and Median House Prices is statistically significant, and we note the need for a better model that would get a better Multiple R-squared value.

```
#
# 3. Age of the house vs Median House Prices
#
ggplot(Boston, aes(age, medv)) +
labs(x="Age of the house", y="Median house prices",
     title = "Age of the house vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```




```
m3 <- lm(medv ~ age, data=Boston)
summary(m3)
```

```
##
## 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
```

Answer - Age of the house vs Median House Prices

The p-value is 2e-16, which indicates a statistically significant association. The Multiple R-squared value is 0.1421 which indicates that our model does a poor job at explaining the response variable. F-statistic value is 83.48 which is far greater than 1, which also indicates a statistically significant association.

Thus we reject the null hypothesis and conclude that relationship between Age of the house and Median House Prices is statistically significant, and we note the need for a better model that would get a better Multiple R-squared value.

```
#
# 4. Weighted mean of distances to five employment centres vs
#Median House Prices
#
ggplot(Boston, aes(dis, medv)) +
labs(x="Weighted mean of distances to five employment centres",
     y="Median house prices",
     title = "Weighted mean of distances to five employment centres vs
     Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```


```
m4 <- lm(medv ~ dis, data=Boston)
summary(m4)
```

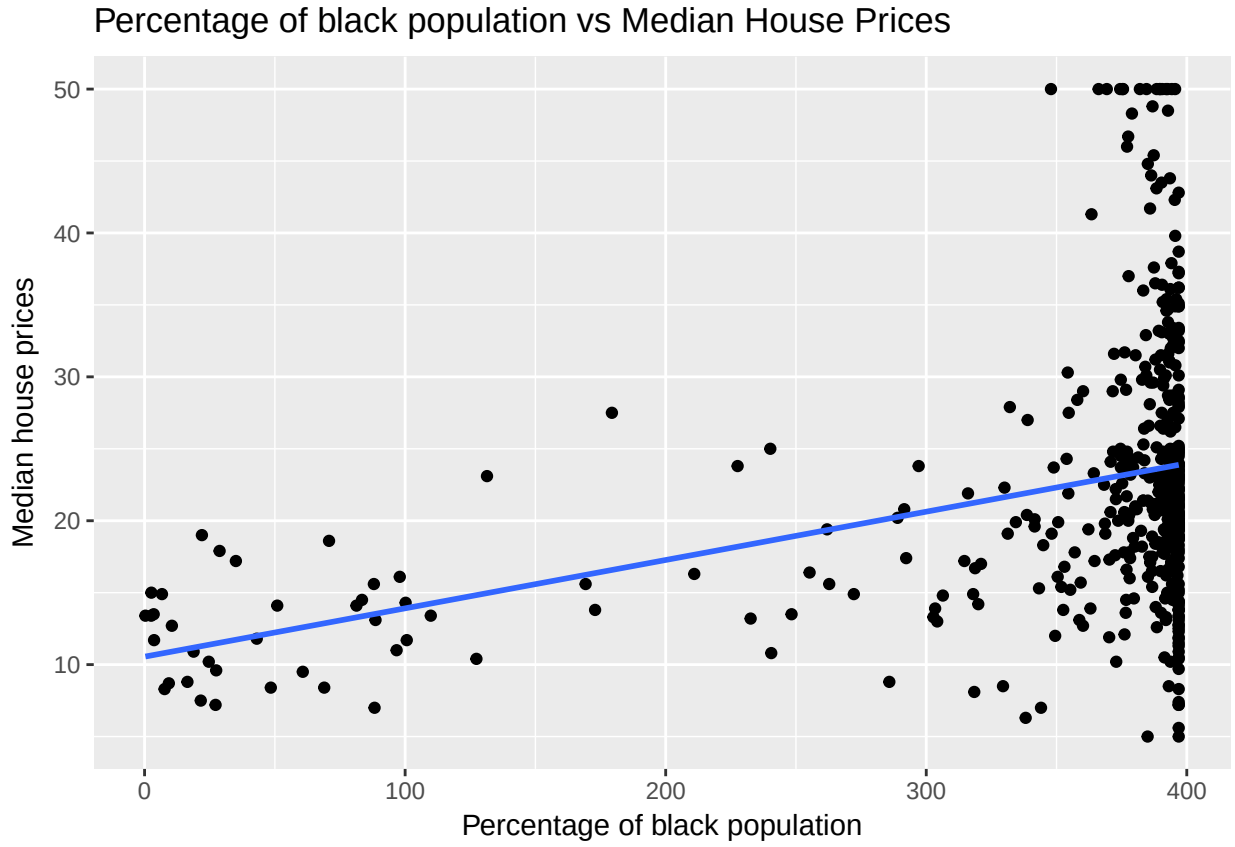
```
##
## Call:
## lm(formula = medv ~ dis, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.016  -5.556  -1.865   2.288  30.377
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.3901     0.8174  22.499  < 2e-16 ***
## dis           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
```

Answer - Distances to five employment centre vs Median House Prices

The p-value is 1.207e-08, which indicates a statistically significant association. The Multiple R-squared value is 0.06246 which indicates that our model does a very poor job at explaining the response variable. F-statistic value is 33.58 which is greater than 1, which also indicates a statistically significant association.

Thus we reject the null hypothesis and conclude that relationship between Weighted mean of distances to five employment centres and Median House Prices is statistically significant, and we note the need for a better model that would get a better Multiple R-squared value.

```
#
# 5. Percentage of black population vs Median House Prices
#
ggplot(Boston, aes(black, medv)) +
labs(x="Percentage of black population", y="Median house prices",
     title = "Percentage of black population vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```




```
m5 <- lm(medv ~ black, data=Boston)
summary(m5)
```

```
##
## 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
```

Answer - Percentage of black population vs Median House Prices

The p-value is 1.318e-14, which indicates a statistically significant association. The Multiple R-squared value is 0.1112 which indicates that our model does a poor job at explaining the response variable. F-statistic value is 63.05 which is greater than 1, which also indicates a statistically significant association.

Thus we reject the null hypothesis and conclude that relationship between Percentage of black population and Median House Prices is statistically significant, and we note the need for a better model that would get a better Multiple R-squared value.

1.4 Multiple Regression

Make sure you are familiar with multiple regression (Openintro Statistics, Ch 8.1-8.3).

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 $H_0 : \beta_j = 0$?

```
#
# Fit a multiple regression model for above 5 predictor variables.
#
multiple.regression <- lm(medv ~ crim + rm + age + dis + black, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ crim + rm + age + dis + black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.907  -2.883  -0.811   1.731  38.597
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.431325   3.193895  -7.649 1.05e-13 ***
## crim         -0.176372   0.034814  -5.066 5.72e-07 ***
## rm            8.008835   0.385660  20.767  < 2e-16 ***
## age          -0.085976   0.014124  -6.087 2.29e-09 ***
## dis          -0.811180   0.190023  -4.269 2.35e-05 ***
## black         0.017504   0.003138   5.578 3.98e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.844 on 500 degrees of freedom
## Multiple R-squared:  0.6002, Adjusted R-squared:  0.5962
## F-statistic: 150.1 on 5 and 500 DF,  p-value: < 2.2e-16
```

Answer -

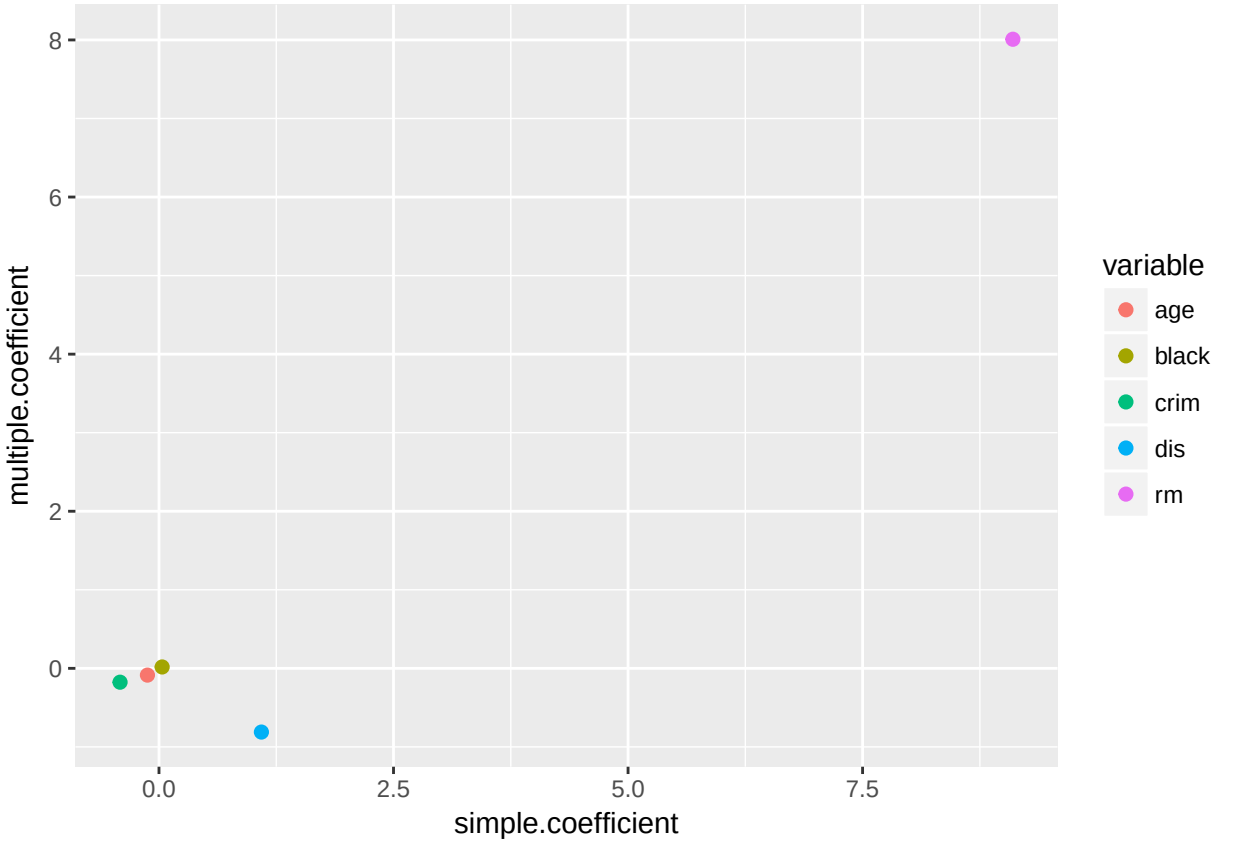
For all 5 of above variables (crime, number of rooms, age of house, distance from employment centers and black population), the p-value is less than 0.05. This means we can reject null hypothesis for all of them, and we can conclude that all 5 variables have a statistically significant relationship with median house value.

Though very simplistic, this is a relatively strong multiple regression model. The F-statistic value is 150.1 which is much better than 1, and Multiple R-squared value is 0.6002 which indicates that our model does a decent job at explaining the response variable which is more than 60% of the times.

1.5 Compare Regressions

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.

```
#
# Compare simple and multiple regressions
#
simple.coefficient <- c(m1$coefficient[2], m2$coefficient[2], m3$coefficient[2],
                        m4$coefficient[2], m5$coefficient[2])
multiple.coefficient <- c(multiple.regression$coefficient[2],
                          multiple.regression$coefficient[3],
                          multiple.regression$coefficient[4],
                          multiple.regression$coefficient[5],
                          multiple.regression$coefficient[6])
variable <- c("crim", "rm", "age", "dis", "black")
d <- data.frame(simple.coefficient, multiple.coefficient, variable)
ggplot(d, aes(simple.coefficient, multiple.coefficient, color=variable)) +
  geom_point(size=2)
```




```
d
```

```
##       simple.coefficient multiple.coefficient variable
## crim          -0.41519028          -0.17637249     crim
## rm             9.10210898           8.00883546       rm
## age           -0.12316272          -0.08597567      age
## dis            1.09161302          -0.81117986      dis
## black          0.03359306           0.01750363    black
```

Answer -

We can compare the values of the coefficients printed above and draw conclusions. For rm, age and black, the coefficients are very similar, but for dis and crim, we can see some variation in the value.

This is due to the fact that for univariate regression, only linear relationship between the specified two variables is considered, other variables are not taken into account.

For multivariate regression, a relationship between two variables is measured by keeping all others constant, and the process continues until all relationships are mapped. So all relationships affect each other, and the values tend to be different from univariate relationship.

1.6 Non-linearities

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:

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

Answer -

We will fit the relationships in a polynomial equation using a combination of lm() and poly function. Also, we will draw some plots with a non-linear equation to see if they are a better fit.

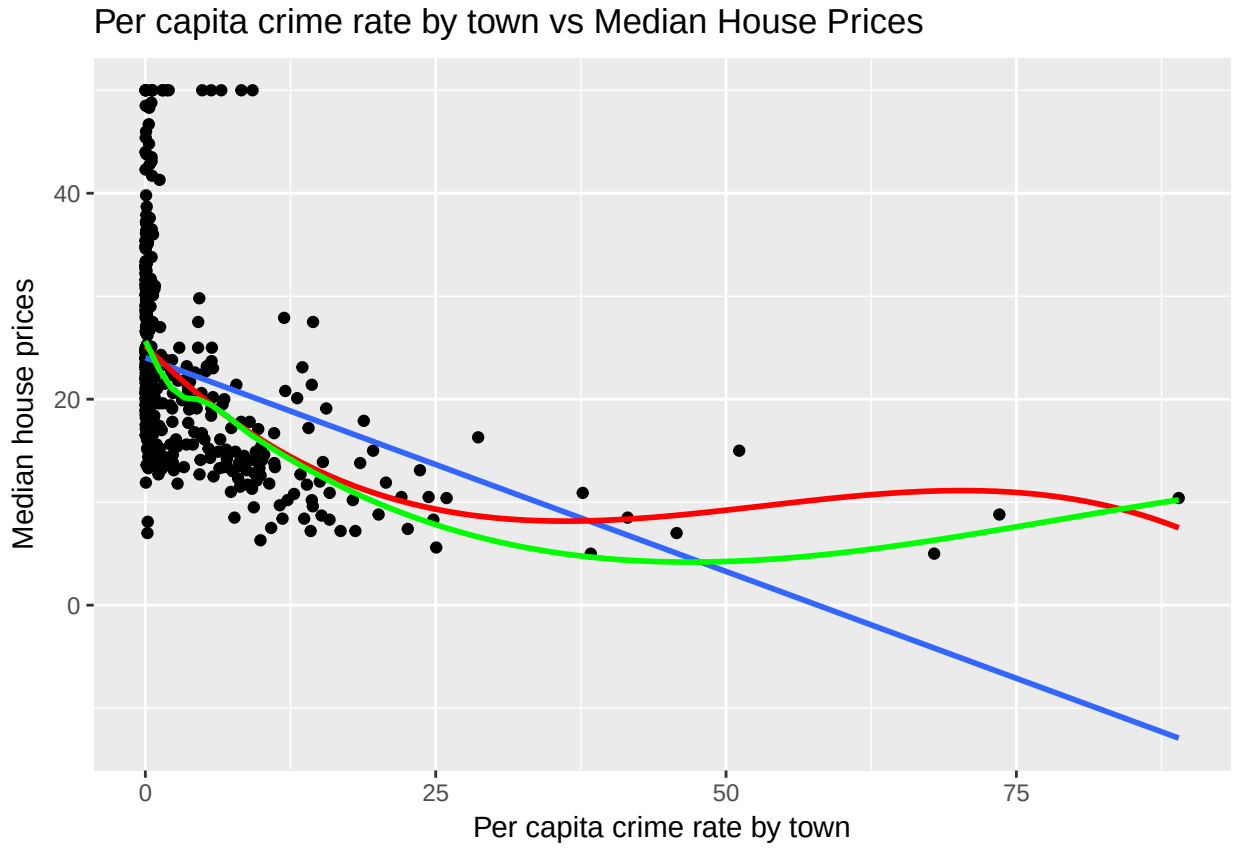
We will draw following three lines/curves -
Blue - linear regression line
Red - Polynomial equation curve
Green - Loess line (best fitting curve)

```
#
# Fit the crime relationship in a polynomial equation.
#
crim.fit <- lm(medv ~ poly(crim, 3), data=Boston)
summary(crim.fit)
```

```
##
## Call:
## lm(formula = medv ~ poly(crim, 3), 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)     22.5328     0.3627  62.124  < 2e-16 ***
## poly(crim, 3)1 -80.2545     8.1589  -9.836  < 2e-16 ***
## poly(crim, 3)2  50.2416     8.1589   6.158 1.51e-09 ***
## poly(crim, 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
```

```
ggplot(Boston, aes(crim, medv)) +
labs(x="Per capita crime rate by town", y="Median house prices",
     title = "Per capita crime rate by town vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE) +
geom_smooth(method = "lm", formula = y ~ poly(x, 3), color="Red", se = FALSE) +
geom_smooth(color="Green", se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```

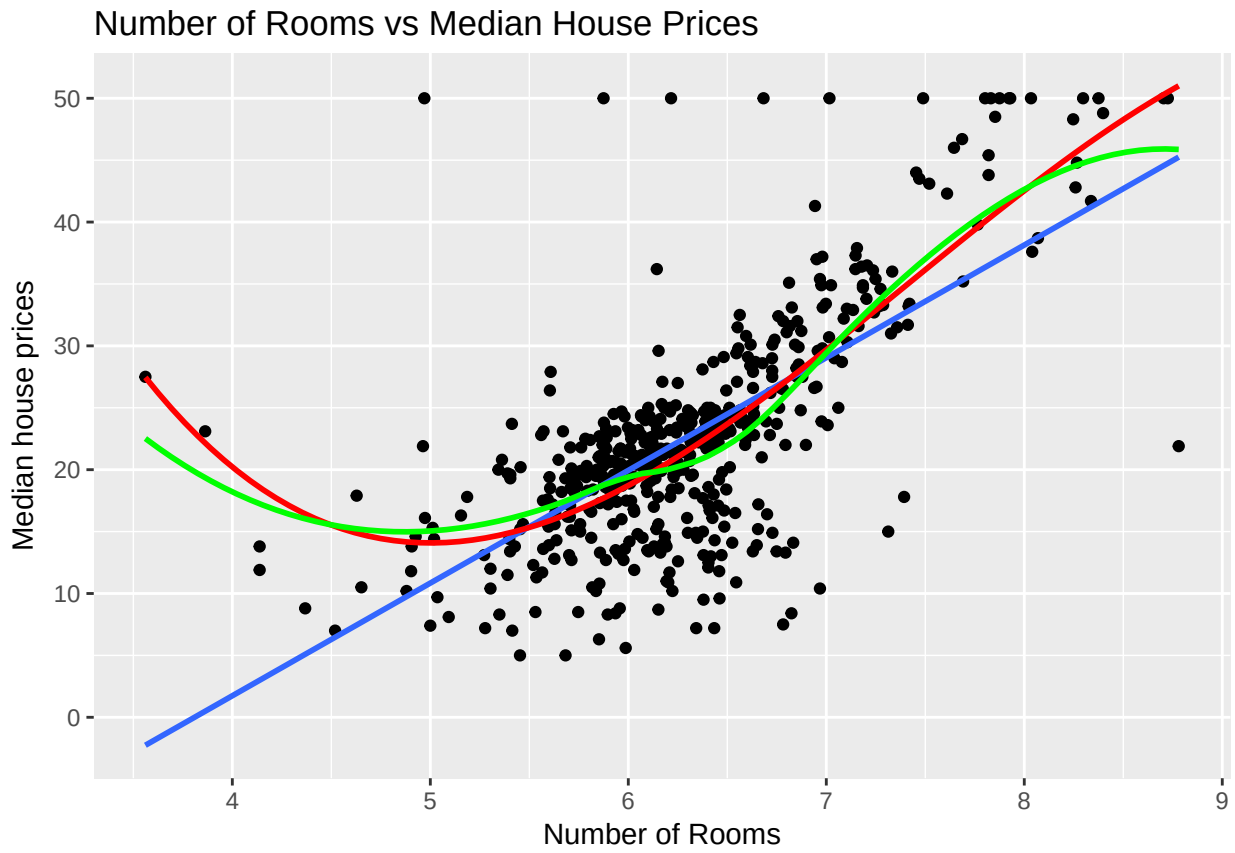



```
#
# Fit the room relationship in a polynomial equation.
#
rm.fit <- lm(medv ~ poly(rm, 3), data=Boston)
summary(rm.fit)
```

```
##
## Call:
## lm(formula = medv ~ poly(rm, 3), 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)   22.5328     0.2716  82.952  < 2e-16 ***
## poly(rm, 3)1 143.7164     6.1103  23.520  < 2e-16 ***
## poly(rm, 3)2  52.6526     6.1103   8.617  < 2e-16 ***
## poly(rm, 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
```

```
ggplot(Boston, aes(rm, medv)) +
labs(x="Number of Rooms", y="Median house prices",
     title = "Number of Rooms vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE) +
geom_smooth(method = "lm", formula = y ~ poly(x, 3), color="Red", se = FALSE) +
geom_smooth(color="Green", se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



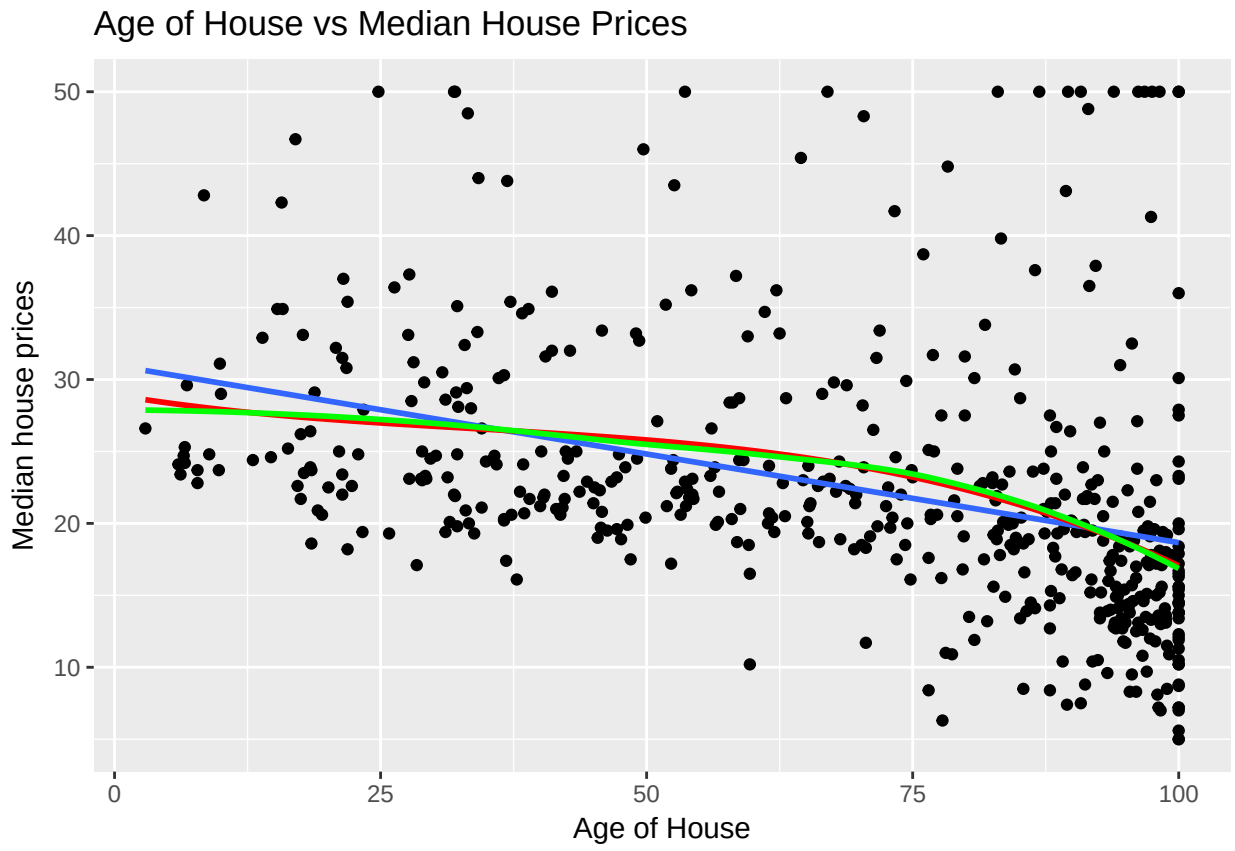

```
#
# Fit the age relationship in a polynomial equation.
#
age.fit <- lm(medv ~ poly(age, 3), data=Boston)
summary(age.fit)
```

```
##
## Call:
## lm(formula = medv ~ poly(age, 3), 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)    22.5328     0.3766  59.830   <2e-16 ***
```

```
## poly(age, 3)1 -77.9087     8.4717  -9.196   <2e-16 ***
## poly(age, 3)2 -23.3290     8.4717  -2.754   0.0061 **
## poly(age, 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
```

```
ggplot(Boston, aes(age, medv)) +
labs(x="Age of House", y="Median house prices",
     title = "Age of House vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE) +
geom_smooth(method = "lm", formula = y ~ poly(x, 3), color="Red", se = FALSE) +
geom_smooth(color="Green", se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



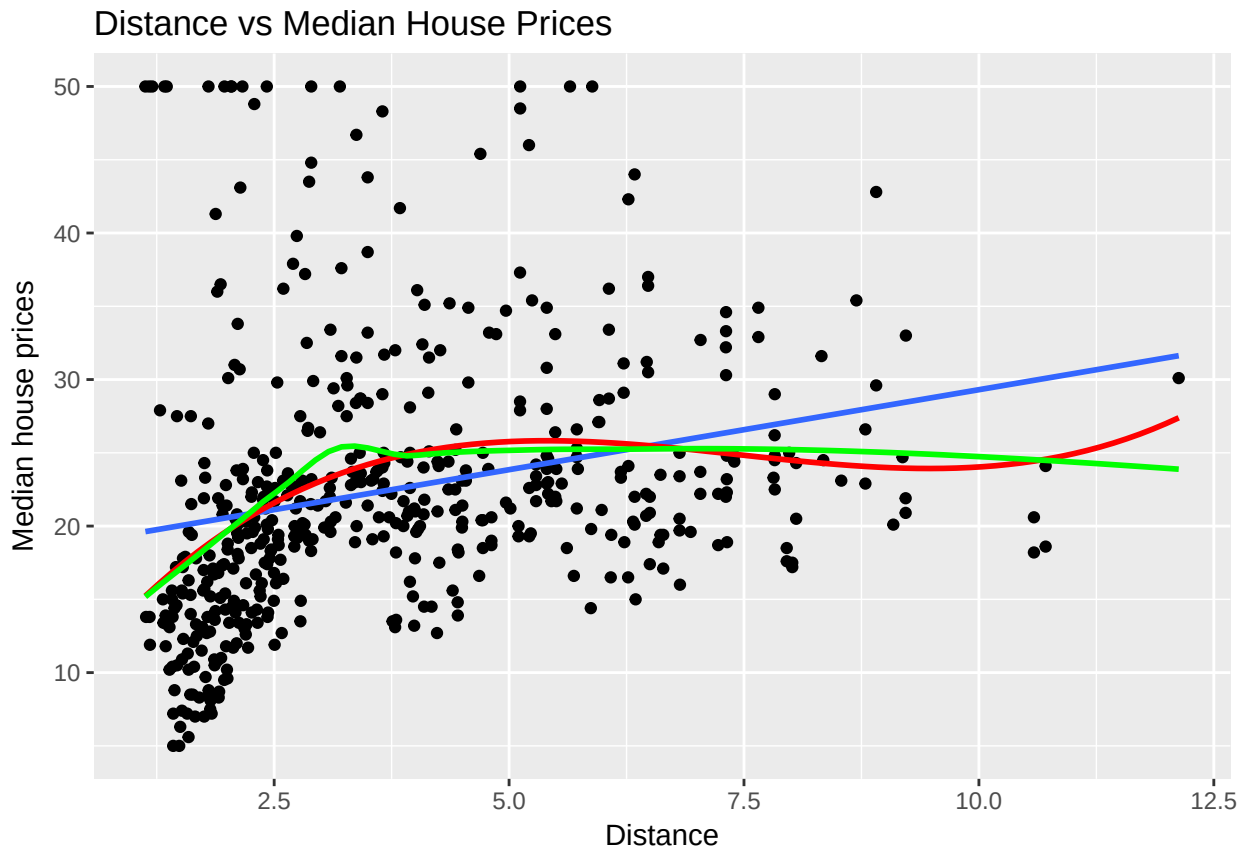

```
#
# Fit the distance relationship in a polynomial equation.
#
dis.fit <- lm(medv ~ poly(dis, 3), data=Boston)
summary(dis.fit)
```

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

```
## lm(formula = medv ~ poly(dis, 3), 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)    22.5328     0.3879  58.082  < 2e-16 ***
## poly(dis, 3)1  51.6551     8.7267   5.919 6.00e-09 ***
## poly(dis, 3)2 -37.5859     8.7267  -4.307 1.99e-05 ***
## poly(dis, 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
```

```
ggplot(Boston, aes(dis, medv)) +
labs(x="Distance", y="Median house prices",
     title = "Distance vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE) +
geom_smooth(method = "lm", formula = y ~ poly(x, 3), color="Red", se = FALSE) +
geom_smooth(color="Green", se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```

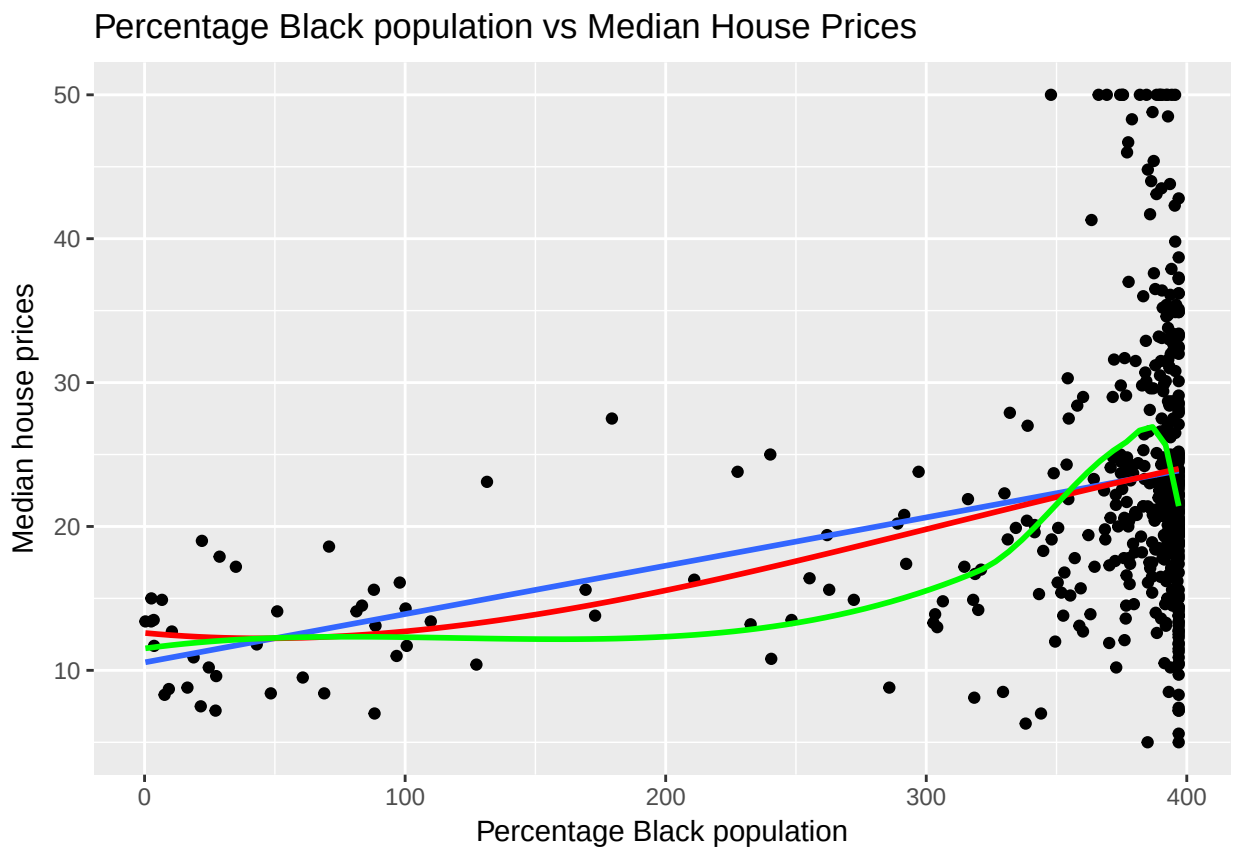



```
#
# Fit the black population relationship in a polynomial equation.
#
black.fit <- lm(medv ~ poly(black, 3), data=Boston)
summary(black.fit)
```

```
##
## Call:
## lm(formula = medv ~ poly(black, 3), 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)      22.5328     0.3861  58.360  < 2e-16 ***
## poly(black, 3)1  68.9194     8.6851   7.935 1.38e-14 ***
## poly(black, 3)2   9.1467     8.6851   1.053    0.293
## poly(black, 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
```

```
ggplot(Boston, aes(black, medv)) +
labs(x="Percentage Black population", y="Median house prices",
     title = "Percentage Black population vs Median House Prices") +
geom_point() +
geom_smooth(method=lm, se=FALSE) +
geom_smooth(method = "lm", formula = y ~ poly(x, 3), color="Red", se = FALSE) +
geom_smooth(color="Green", se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



From this we can conclude that for all 5 predictor variables, the relationship with medv is not perfectly linear. However, for age of the house, the relationship is very close to linear.

1.7 Stepwise Model Selection

Consider performing a stepwise model selection procedure to determine the best fit model (consult Openintro Statistics, 8.2.2). Discuss your results. How is this model different from the model in (4)?

Answer -

We will use 'Backward Elimination' strategy and start eliminating predictors from 1.4 and veryfy the results. Our original R square is 0.6002 so if we get a better value, we will adopt that model.

```
#
# Original Regression
#
```

```r
multiple.regression <- lm(medv ~ crim + rm + age + dis + black, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ crim + rm + age + dis + black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.907  -2.883  -0.811   1.731  38.597
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.431325   3.193895  -7.649 1.05e-13 ***
## crim         -0.176372   0.034814  -5.066 5.72e-07 ***
## rm            8.008835   0.385660  20.767  < 2e-16 ***
## age          -0.085976   0.014124  -6.087 2.29e-09 ***
## dis          -0.811180   0.190023  -4.269 2.35e-05 ***
## black         0.017504   0.003138   5.578 3.98e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.844 on 500 degrees of freedom
## Multiple R-squared:  0.6002, Adjusted R-squared:  0.5962
## F-statistic: 150.1 on 5 and 500 DF,  p-value: < 2.2e-16
```

```r
#
# Drop crim
#
multiple.regression <- lm(medv ~ rm + age + dis + black, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ rm + age + dis + black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.164  -2.598  -0.578   1.838  39.136
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.554462   3.163570  -9.026  < 2e-16 ***
## rm            8.264589   0.391641  21.102  < 2e-16 ***
## age          -0.091195   0.014429  -6.320 5.78e-10 ***
## dis          -0.667231   0.192455  -3.467 0.000572 ***
## black         0.022242   0.003068   7.250 1.59e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.986 on 501 degrees of freedom
## Multiple R-squared:  0.5797, Adjusted R-squared:  0.5763
## F-statistic: 172.7 on 4 and 501 DF,  p-value: < 2.2e-16
```

```
#
# Drop rm
#
multiple.regression <- lm(medv ~ crim + age + dis + black, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ crim + age + dis + black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.831  -4.877  -2.254   1.860  30.573
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.979322   2.668655  10.484  < 2e-16 ***
## crim        -0.271011   0.047057  -5.759 1.48e-08 ***
## age         -0.120298   0.019125  -6.290 6.92e-10 ***
## dis         -0.768799   0.259056  -2.968  0.00314 **
## black        0.018784   0.004277   4.392 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.968 on 501 degrees of freedom
## Multiple R-squared:  0.2554, Adjusted R-squared:  0.2495
## F-statistic: 42.96 on 4 and 501 DF,  p-value: < 2.2e-16
```

```
#
# Drop age
#
multiple.regression <- lm(medv ~ crim + rm + dis + black, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ crim + rm + dis + black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.304  -2.909  -0.676   2.380  38.237
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -35.410284   2.729041 -12.975  < 2e-16 ***
## crim         -0.191830   0.035949  -5.336 1.44e-07 ***
## rm            8.283531   0.396552  20.889  < 2e-16 ***
## dis          -0.007234   0.141448  -0.051    0.959
## black         0.018518   0.003244   5.708 1.96e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.051 on 501 degrees of freedom
## Multiple R-squared:  0.5706, Adjusted R-squared:  0.5672
## F-statistic: 166.4 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
#
# Drop dis
#
multiple.regression <- lm(medv ~ crim + rm + age + black, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ crim + rm + age + black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.929  -3.010  -0.853   1.911  39.246
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -29.961964   2.969079 -10.091  < 2e-16 ***
## crim         -0.154149   0.035010  -4.403 1.31e-05 ***
## rm            7.991154   0.392211  20.375  < 2e-16 ***
## age          -0.044067   0.010328  -4.267 2.37e-05 ***
## black         0.016408   0.003181   5.159 3.59e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.944 on 501 degrees of freedom
## Multiple R-squared:  0.5856, Adjusted R-squared:  0.5823
## F-statistic:   177 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
#
# Drop black
#
multiple.regression <- lm(medv ~ crim + rm + age + dis, data=Boston)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = medv ~ crim + rm + age + dis, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.834  -2.658  -0.631   1.881  39.402
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.28704    3.08675  -5.924 5.83e-09 ***
## crim         -0.23426    0.03422  -6.847 2.22e-11 ***
## rm            8.05112    0.39701  20.280  < 2e-16 ***
## age          -0.09016    0.01452  -6.208 1.12e-09 ***
## dis          -0.72445    0.19499  -3.715 0.000226 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.017 on 501 degrees of freedom
## Multiple R-squared:  0.5753, Adjusted R-squared:  0.5719
## F-statistic: 169.7 on 4 and 501 DF,  p-value: < 2.2e-16
```

As can be seen, none of the new models provide a better R square value than the original one. We can do more trial and error, and drop more than one predictors, and keep trying until we get a better value. We can also try to fit in better non-linear models to better explain the response variable. But for the purpose of this exercise, the original model from 1.4 gives us the best R square value.

1.8 Do Assumptions Hold?

Evaluate the statistical assumptions in your regression analysis from (1.7) by performing a basic analysis of model residuals and any unusual observations (consult Openintro Statistics 7.2). Discuss any concerns you have about your model.

Answer -

We will comment on some models we made in 1.7 with Backward Elimination method.

Multiple R-squared values with univariate regressions -
crim - 0.1508
rm - 0.4835
age - 0.1421
dis - 0.06246
black - 0.1112

Multiple R-squared values with multivariate regressions -
multiple regression with all 5 predictor variables included - 0.6002
rm + age + dis + black - 0.5797
crim + age + dis + black - 0.2554
crim + rm + dis + black - 0.5706
crim + rm + age + black - 0.5856
crim + rm + age + dis - 0.5753

It can be observed that when 'rm' was dropped, which is the most powerful model in univariate regressions, the R square value for crim + age + dis + black dropped significantly, and that is as expected.

For all other combinations, the R square values are very similar, and do not represent any significant change. Looking at univariate values, it is acceptable and expected.

The coefficient analysis has been done in 1.5, and there is not much more to add. All coefficients vary in an acceptable range and nammer. I noted only one concern. The coeffiient of dis dropped significantly when age was excluded from the analysis. It would be interesting to investigate that reason.

Also I have one other outstanding question. With all 5 predictors included, we get an R square value of 0.6002. In univariate regressions, none of the values are that strong. It would be an interesting exercise to find out why.

2. Diamonds' Price

Let's look at the *diamonds* dataset from *ggplot2* package. Your task is to find which parameters influence the price of diamonds.

I recommend to transform the ordered factors (such as *cut*, *clarity*) to unordered factors with a command like `factor(cut, ordered=FALSE)` in order to give more easily interpretable results.

2.1 Describe the variables.

What do you think, which variables are relevant in determining the price? Describe your thought before you do any formal analysis.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  1.3.4     v purrr   0.2.4
## v tidyr   0.7.2     v dplyr   0.7.4
## v readr   1.1.1     v stringr 1.2.0
## v tibble  1.3.4     v forcats 0.2.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x dplyr::select()    masks MASS::select()
## x purrr::transpose() masks data.table::transpose()
```

```
data(diamonds)
as.data.table(diamonds)
```

```
##        carat       cut color clarity depth table price    x    y    z
##     1:  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
##     2:  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
##     3:  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
##     4:  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
##     5:  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
##    ---
## 53936:  0.72     Ideal     D     SI1  60.8    57  2757 5.75 5.76 3.50
## 53937:  0.72      Good     D     SI1  63.1    55  2757 5.69 5.75 3.61
## 53938:  0.70 Very Good     D     SI1  62.8    60  2757 5.66 5.68 3.56
## 53939:  0.86   Premium     H     SI2  61.0    58  2757 6.15 6.12 3.74
## 53940:  0.75     Ideal     D     SI2  62.2    55  2757 5.83 5.87 3.64
```

```
diamonds$cut = factor(diamonds$cut, ordered = FALSE)
diamonds$color = factor(diamonds$color, ordered = FALSE)
diamonds$clarity = factor(diamonds$clarity, ordered = FALSE)
```

Answer -

This is a data frame with 53940 rows and 10 variables with the prices and other attributes of individual diamonds.

The variables are as follows:

price - price in US dollars ($326-$18,823)
carat - weight of the diamond (0.2-5.01)
cut - quality of the cut (Fair, Good, Very Good, Premium, Ideal)
color - diamond colour, from J (worst) to D (best)
clarity - a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
x - length in mm (0-10.74)
y - width in mm (0-58.9)
z - depth in mm (0-31.8)
depth - total depth percentage = z /mean(x, y) = 2 * z / (x + y) (43-79)
table - width of top of diamond relative to widest point (43-95)

Without doing any analysis, it seems that following 4 variables will influence diamond price -
carat

cut
color
clarity.

2.2 Multiple regression

Select a number of variables you consider the most relevant. Estimate a multiple regression model in the form

$$\text{price}_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \epsilon_i.$$

Interpret the coefficient values.

- if you are able to, give the literal interpretation of the numeric value
- if there is no easy literal interpretation, broadly explain what it means, and interpret at least the sign.

```
md <- lm(price ~ carat + cut + color + clarity, data = diamonds)
summary(md)
```

```
##
## Call:
## lm(formula = price ~ carat + cut + color + clarity, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16813.5   -680.4   -197.6    466.4  10394.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7362.80      51.68 -142.46   <2e-16 ***
## carat         8886.13      12.03  738.44   <2e-16 ***
## cutGood        655.77      33.63   19.50   <2e-16 ***
## cutVery Good   848.72      31.28   27.14   <2e-16 ***
## cutPremium     869.40      30.93   28.11   <2e-16 ***
## cutIdeal       998.25      30.66   32.56   <2e-16 ***
## colorE        -211.68      18.32  -11.56   <2e-16 ***
## colorF        -303.31      18.51  -16.39   <2e-16 ***
## colorG        -506.20      18.12  -27.93   <2e-16 ***
## colorH        -978.70      19.27  -50.78   <2e-16 ***
## colorI       -1440.30      21.65  -66.54   <2e-16 ***
## colorJ       -2325.22      26.72  -87.01   <2e-16 ***
## claritySI2    2625.95      44.79   58.63   <2e-16 ***
## claritySI1    3573.69      44.60   80.13   <2e-16 ***
## clarityVS2    4217.83      44.84   94.06   <2e-16 ***
## clarityVS1    4534.88      45.54   99.59   <2e-16 ***
## clarityVVS2   4967.20      46.89  105.93   <2e-16 ***
## clarityVVS1   5072.03      48.21  105.20   <2e-16 ***
## clarityIF     5419.65      52.14  103.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1157 on 53921 degrees of freedom
## Multiple R-squared:  0.9159, Adjusted R-squared:  0.9159
## F-statistic: 3.264e+04 on 18 and 53921 DF,  p-value: < 2.2e-16
```

Answer -

The intercept is -7362.80. Broadly speaking, when all predictor variables are zero, this is the price of the diamond. So a price of a diamond with 0 carat, no cut, no color and no clarity is -7362.80.

There is no easy explanation as to why it is negative, and such a large value. I will try to provide a high level explanation of this.

Coefficients in a multivariate regression indicate change in response variable, when one predictor variable is increased by 1 unit, keeping all other predictor variables at constant value. Looking at numbers, carat has a huge impact on price. Each additional carat causes a price increase of 8886.13. Similarly for clarity, 1 unit increase tends to drastically increase diamond price. Considering such a large impact on the price, when all of these are suddenly reduced to zero, the price sharply drops. So sharply that it goes well below zero and achieves a value of -7362.80.

One way to explain it could be look at the diamond prices with lowest carat. They are at about 400 in price with a carat weight of 0.2. When carat, and everything else reduced to zero, the price sharply drops. Carat in itself causes a price reduction of about 1700, clarity causes a reduction of about 3500 and then cut and color cause further reduction. This in effect takes the intercept well below zero, at -7362.80.

2.3 Other specifications

Select 2-3 different sets of explanatory variables or change the model specification in other ways, for instance by using log of the outcome or explanatory variables, adding interactions and squares, cubes of the variables, normalizing variables, or something else.

Which specification gives you the highest R^2? Comment your results.

Answer -

We will try multiple combinations of multivariate regressions and look at R^2 values.

```
#
# Regression 1
#
r1 <- lm(log(price) ~ carat + cut + color + clarity, data = diamonds)
summary(r1)
```

```
##
## Call:
## lm(formula = log(price) ~ carat + cut + color + clarity, data = diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9828 -0.2183  0.0576  0.2485  1.6301
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.353040   0.015141 353.553  < 2e-16 ***
## carat         2.197459   0.003525 623.348  < 2e-16 ***
## cutGood       0.050432   0.009853   5.118 3.09e-07 ***
## cutVery Good  0.058855   0.009163   6.423 1.34e-10 ***
## cutPremium    0.056908   0.009061   6.280 3.40e-10 ***
## cutIdeal      0.084137   0.008981   9.369  < 2e-16 ***
## colorE       -0.055664   0.005366 -10.374  < 2e-16 ***
## colorF       -0.052365   0.005422  -9.657  < 2e-16 ***
## colorG       -0.128693   0.005309 -24.241  < 2e-16 ***
## colorH       -0.261627   0.005646 -46.341  < 2e-16 ***
```

```
## colorI       -0.418334   0.006341 -65.970  < 2e-16 ***
## colorJ       -0.580406   0.007828 -74.142  < 2e-16 ***
## claritySI2    0.541690   0.013121  41.285  < 2e-16 ***
## claritySI1    0.725088   0.013065  55.499  < 2e-16 ***
## clarityVS2    0.819252   0.013136  62.366  < 2e-16 ***
## clarityVS1    0.882720   0.013339  66.174  < 2e-16 ***
## clarityVVS2   0.931597   0.013737  67.819  < 2e-16 ***
## clarityVVS1   0.941856   0.014123  66.688  < 2e-16 ***
## clarityIF     1.025866   0.015273  67.168  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3389 on 53921 degrees of freedom
## Multiple R-squared:  0.8885, Adjusted R-squared:  0.8884
## F-statistic: 2.387e+04 on 18 and 53921 DF,  p-value: < 2.2e-16
```

```
#
# Regression 2
#
r2 <- lm(price^2 ~ carat + cut + color + clarity, data = diamonds)
summary(r2)
```

```
##
## Call:
## lm(formula = price^2 ~ carat + cut + color + clarity, data = diamonds)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -187725175  -21913919   -4636248   14457905  248504680
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -131665308    1420346 -92.699  < 2e-16 ***
## carat         120731101     330703 365.074  < 2e-16 ***
## cutGood        10831200     924311  11.718  < 2e-16 ***
## cutVery Good   14476141     859538  16.842  < 2e-16 ***
## cutPremium     15598714     850029  18.351  < 2e-16 ***
## cutIdeal       18036764     842475  21.409  < 2e-16 ***
## colorE         -2660367     503346  -5.285 1.26e-07 ***
## colorF         -5380907     508667 -10.578  < 2e-16 ***
## colorG         -8218871     498018 -16.503  < 2e-16 ***
## colorH        -14872816     529620 -28.082  < 2e-16 ***
## colorI        -19066244     594870 -32.051  < 2e-16 ***
## colorJ        -34008305     734374 -46.309  < 2e-16 ***
## claritySI2     39676018    1230849  32.235  < 2e-16 ***
## claritySI1     53560507    1225607  43.701  < 2e-16 ***
## clarityVS2     64437250    1232306  52.290  < 2e-16 ***
## clarityVS1     69089442    1251372  55.211  < 2e-16 ***
## clarityVVS2    76907775    1288621  59.682  < 2e-16 ***
## clarityVVS1    80701661    1324903  60.911  < 2e-16 ***
## clarityIF      86929907    1432776  60.672  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31790000 on 53921 degrees of freedom
```

```
## Multiple R-squared:  0.7239, Adjusted R-squared:  0.7238
## F-statistic:  7853 on 18 and 53921 DF,  p-value: < 2.2e-16
```

```
#
# Regression 3
#
r3 <- lm(price^0.5 ~ carat + cut + color + clarity, data = diamonds)
summary(r3)
```

```
##
## Call:
## lm(formula = price^0.5 ~ carat + cut + color + clarity, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.936   -3.296   -0.478    2.824   51.690
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -21.63307    0.28602  -75.64   <2e-16 ***
## carat         64.55472    0.06659  969.37   <2e-16 ***
## cutGood        3.49303    0.18613   18.77   <2e-16 ***
## cutVery Good   4.38902    0.17309   25.36   <2e-16 ***
## cutPremium     4.31129    0.17117   25.19   <2e-16 ***
## cutIdeal       5.12887    0.16965   30.23   <2e-16 ***
## colorE        -1.58712    0.10136  -15.66   <2e-16 ***
## colorF        -1.81954    0.10243  -17.76   <2e-16 ***
## colorG        -3.54749    0.10029  -35.37   <2e-16 ***
## colorH        -7.05349    0.10665  -66.14   <2e-16 ***
## colorI       -11.03506    0.11979  -92.12   <2e-16 ***
## colorJ       -16.61846    0.14788 -112.38   <2e-16 ***
## claritySI2    17.62987    0.24786   71.13   <2e-16 ***
## claritySI1    23.82917    0.24680   96.55   <2e-16 ***
## clarityVS2    27.56491    0.24815  111.08   <2e-16 ***
## clarityVS1    29.66552    0.25199  117.72   <2e-16 ***
## clarityVVS2   31.95572    0.25949  123.15   <2e-16 ***
## clarityVVS1   32.27949    0.26680  120.99   <2e-16 ***
## clarityIF     34.58665    0.28852  119.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.402 on 53921 degrees of freedom
## Multiple R-squared:  0.9501, Adjusted R-squared:   0.95
## F-statistic: 5.7e+04 on 18 and 53921 DF,  p-value: < 2.2e-16
```

```
#
# Regression 4
#
r4 <- lm(price^0.5 ~ carat + cut + clarity, data = diamonds)
summary(r4)
```

```
##
## Call:
## lm(formula = price^0.5 ~ carat + cut + clarity, data = diamonds)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -150.761   -3.901   -0.526    3.237   57.743
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -23.11321    0.32831  -70.40   <2e-16 ***
## carat         61.49806    0.07471  823.18   <2e-16 ***
## cutGood        3.59028    0.22050   16.28   <2e-16 ***
## cutVery Good   4.52078    0.20505   22.05   <2e-16 ***
## cutPremium     4.40697    0.20280   21.73   <2e-16 ***
## cutIdeal       5.27709    0.20098   26.26   <2e-16 ***
## claritySI2    17.91894    0.29360   61.03   <2e-16 ***
## claritySI1    23.27692    0.29231   79.63   <2e-16 ***
## clarityVS2    26.94244    0.29393   91.66   <2e-16 ***
## clarityVS1    28.34783    0.29830   95.03   <2e-16 ***
## clarityVVS2   31.38551    0.30732  102.13   <2e-16 ***
## clarityVVS1   31.00850    0.31584   98.18   <2e-16 ***
## clarityIF     33.06020    0.34114   96.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.586 on 53927 degrees of freedom
## Multiple R-squared:  0.9299, Adjusted R-squared:  0.9299
## F-statistic: 5.96e+04 on 12 and 53927 DF,  p-value: < 2.2e-16
```

```
#
# Regression 5
#
r5 <- lm(price^0.5 ~ carat + color + clarity, data = diamonds)
summary(r5)
```

```
##
## Call:
## lm(formula = price^0.5 ~ carat + color + clarity, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -153.428   -3.339   -0.489    2.866   51.312
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.23301    0.26073  -69.93   <2e-16 ***
## carat        64.38941    0.06684  963.27   <2e-16 ***
## colorE       -1.61193    0.10233  -15.75   <2e-16 ***
## colorF       -1.88041    0.10341  -18.18   <2e-16 ***
## colorG       -3.56889    0.10124  -35.25   <2e-16 ***
## colorH       -7.09273    0.10766  -65.88   <2e-16 ***
## colorI      -11.04309    0.12097  -91.29   <2e-16 ***
## colorJ      -16.69850    0.14930 -111.84   <2e-16 ***
## claritySI2   18.68381    0.24731   75.55   <2e-16 ***
## claritySI1   24.96597    0.24579  101.57   <2e-16 ***
## clarityVS2   28.83082    0.24687  116.79   <2e-16 ***
## clarityVS1   30.94991    0.25076  123.42   <2e-16 ***
## clarityVVS2  33.33013    0.25809  129.14   <2e-16 ***
## clarityVVS1  33.71811    0.26533  127.08   <2e-16 ***
```

```
## clarityIF    36.12537    0.28727  125.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.465 on 53925 degrees of freedom
## Multiple R-squared:  0.9491, Adjusted R-squared:  0.9491
## F-statistic: 7.179e+04 on 14 and 53925 DF,  p-value: < 2.2e-16
```

```
#
# Regression 6
#
r6 <-  lm(log(price) ~ log(carat) + cut + clarity + color, data=diamonds)
summary(r6)
```

```
##
## Call:
## lm(formula = log(price) ~ log(carat) + cut + clarity + color,
##     data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01107 -0.08636 -0.00023  0.08341  1.94778
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.856856   0.005758 1364.43   <2e-16 ***
## log(carat)    1.883718   0.001129 1668.75   <2e-16 ***
## cutGood       0.080048   0.003890   20.57   <2e-16 ***
## cutVery Good  0.117215   0.003619   32.39   <2e-16 ***
## cutPremium    0.139345   0.003579   38.94   <2e-16 ***
## cutIdeal      0.161218   0.003548   45.44   <2e-16 ***
## claritySI2    0.427879   0.005178   82.64   <2e-16 ***
## claritySI1    0.592954   0.005149  115.17   <2e-16 ***
## clarityVS2    0.742164   0.005178  143.34   <2e-16 ***
## clarityVS1    0.812277   0.005257  154.52   <2e-16 ***
## clarityVVS2   0.947271   0.005418  174.83   <2e-16 ***
## clarityVVS1   1.018743   0.005575  182.73   <2e-16 ***
## clarityIF     1.113732   0.006030  184.69   <2e-16 ***
## colorE       -0.054277   0.002118  -25.62   <2e-16 ***
## colorF       -0.094596   0.002142  -44.16   <2e-16 ***
## colorG       -0.160378   0.002097  -76.49   <2e-16 ***
## colorH       -0.251071   0.002225 -112.85   <2e-16 ***
## colorI       -0.372574   0.002492 -149.50   <2e-16 ***
## colorJ       -0.510983   0.003074 -166.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1338 on 53921 degrees of freedom
## Multiple R-squared:  0.9826, Adjusted R-squared:  0.9826
## F-statistic: 1.693e+05 on 18 and 53921 DF,  p-value: < 2.2e-16
```

Looking at above results, the Regression # 6 has a R^2 value of 0.9826 which is the best in the attempted models.

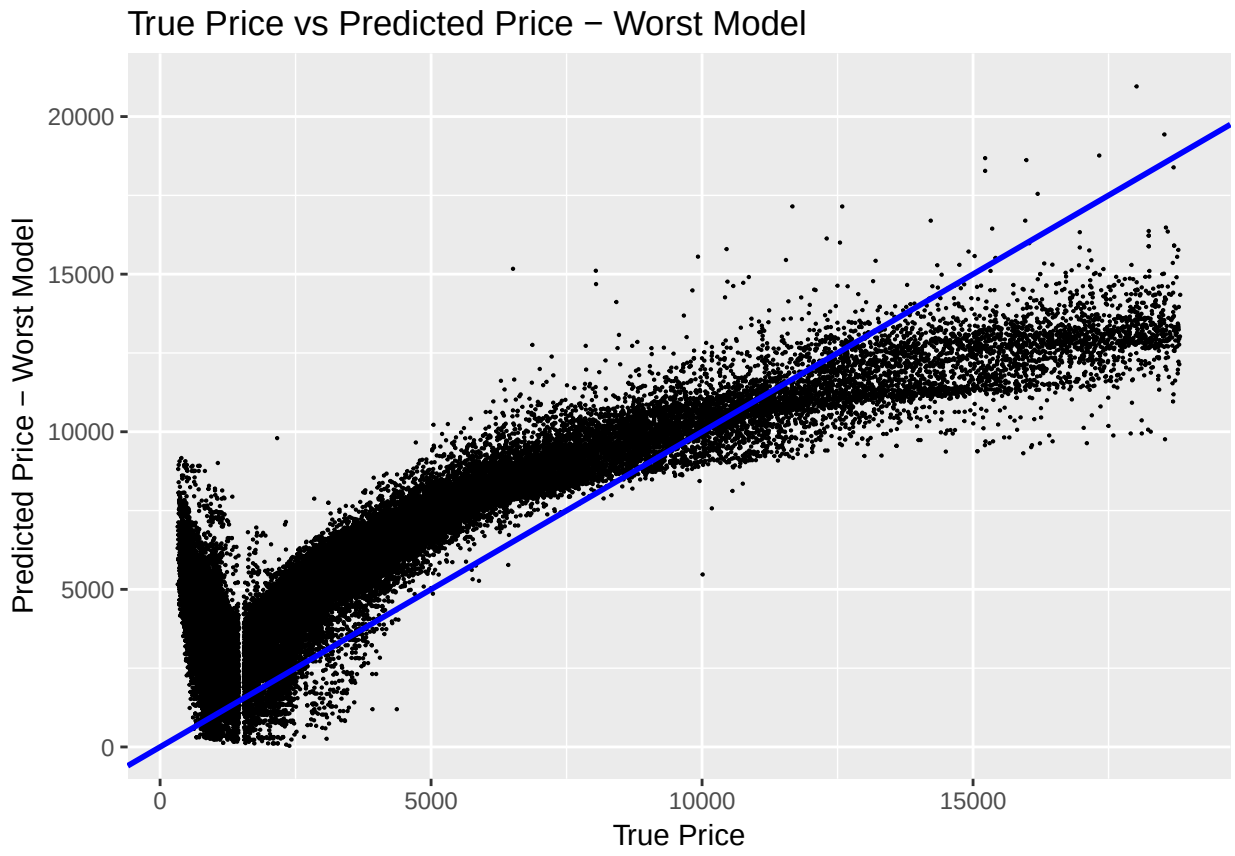
2.4 Visualize your best model

Visualize your best and your worst model's predictions on a true-predicted price scatterplot. Explain the differences.

Answer -

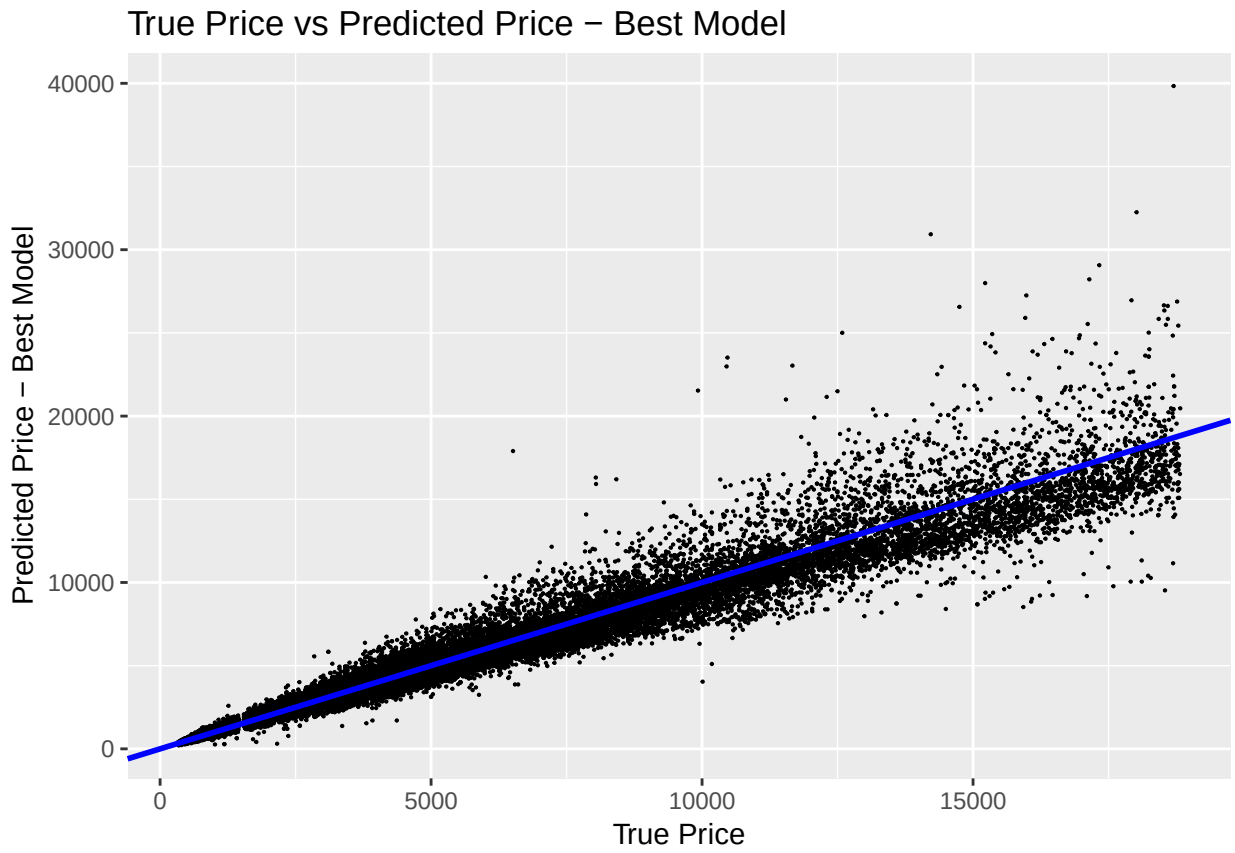
r2 is the worst model at $R^2 = 0.7239$, while r6 is best at $R^2 = 0.9826$. We will predict and plot these, and compare the results.

```
#predict worst model, r2
diamonds$pred.worst <- predict(r2, diamonds)
diamonds$pred.worst <- abs(diamonds$pred.worst)^0.5
#predict best model, r6
diamonds$pred.best <- predict(r6, diamonds)
diamonds$pred.best <- exp(diamonds$pred.best)
#Plot worst model - True vs Predicted
ggplot(diamonds, aes(price, pred.worst)) +
labs(x="True Price", y="Predicted Price - Worst Model",
     title = "True Price vs Predicted Price - Worst Model") +
geom_point(size=0.1) +
geom_abline(intercept=0, slope=1, color="Blue", size=1)
```




```
#Plot best model - True vs Predicted
ggplot(diamonds, aes(price, pred.best)) +
labs(x="True Price", y="Predicted Price - Best Model",
     title = "True Price vs Predicted Price - Best Model") +
geom_point(size=0.1) +
```

```
geom_abline(intercept=0, slope=1, color="Blue", size=1)
```



For a perfect model, all points will fall on the x=y line that is the blue line on the plots. However since our models are not perfect, many points fall outside of the x=y line.

Worst model -

As can be seen, a large number of points fall outside of the x=y line. This indicates that there is a larger variance between values predicted by the model and true values. This is also reflected by the model's weaker R^2 value of 0.7239.

Best Model -

As can be seen, a large number of points fall on the x=y line. Also, there is a large number of points very close to that line. This indicates that the model is superior and provides a better match between predicted values and actual values.

2.5 Residuals

- Show the distribution of residuals (difference between the actual and predicted price). Does it look normal?
- Analyze a few largest outliers. Anything special with those diamonds?

Answer -

We will only comment on the best model. Because of the nature of the questions, it does not make sense to comment on the worst model. That model is weak by definition and the questions don't apply.

We will calculate residual for best model, and plot a histogram. We will then comment on the results.

```r
#calculate residuals and plot a histogram.
residual <- resid(r6)
hist(residual, breaks=150)
m <- mean(residual)
m
```

```
## [1] -6.253612e-17
```

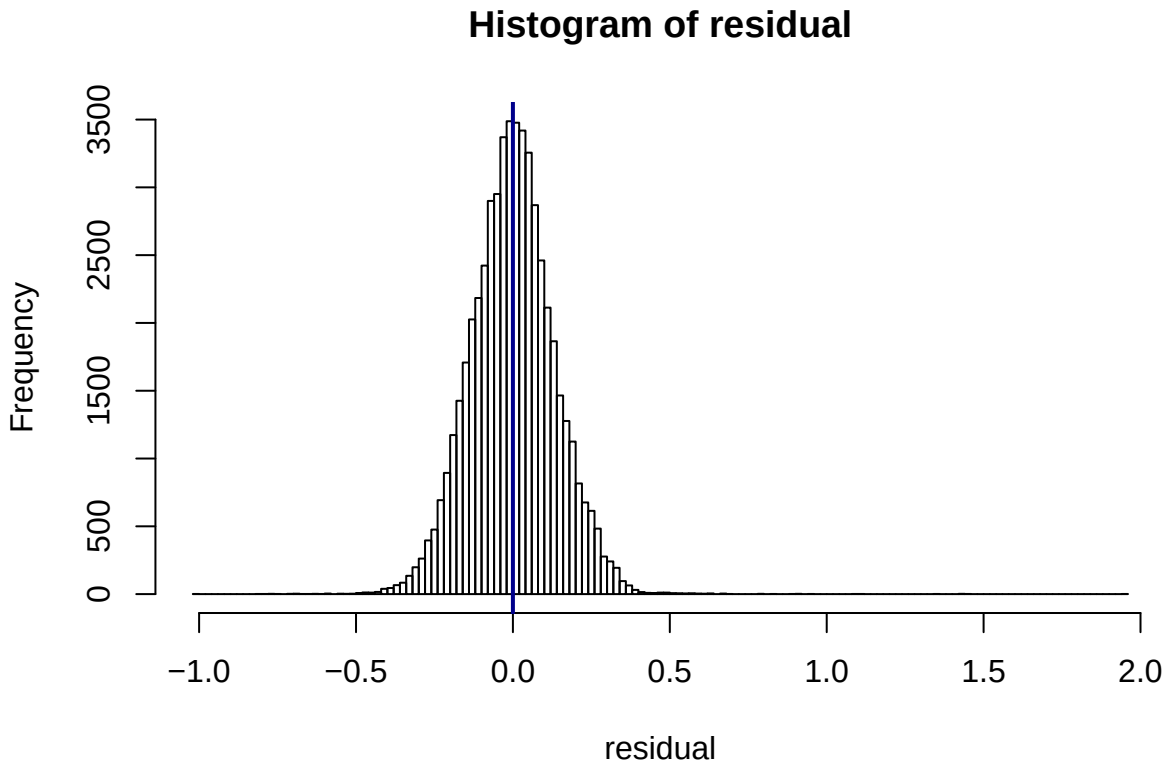
```r
sd <- sd(residual)
sd
```

```
## [1] 0.1337794
```

```r
median <- median(residual)
median
```

```
## [1] -0.0002277932
```

```r
# place the mean on the histogram
abline(v=mean(residual), col="darkblue", lwd=2)
```



As can be seen, the mean is very close to 0. The distribution does look very close to normal.

Outliers -

From the outlier analysis, it can be seen that a diamond is more likely to be an outlier if it is an expensive one. The model calculates higher prices for diamonds with more carat and premium cut etc, but the diamond is not actually worth that much price. This could be due to any factors that our model does not consider, like table, or depth etc. It could be possible that some combinations of attributes is not attractive to buyers, so the diamonds are are worth less than the model predicts. It could be also due to a factor that customers are

not willing to pay beyond a limit for certain attributes, e.g. cut or clarity once it reaches a certain amount.

3. How much work?

Tell us, roughly how many hours did you spend on this homework.

Answer -

Anticipated - 12 hours

Actual - 38 hours