ISLR - Ch3 - Linear Regression

Darshan Patel

1/19/2019

The following set of problems are from the applied exercises section in ISLR Chapter 3: Linear Regression.

```r
library(MASS)
library(ISLR)
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 3.4.4

## Warning: package 'purrr' was built under R version 3.4.4

## Warning: package 'dplyr' was built under R version 3.4.4
```

Question 8: This question involves the use of simple linear regression on the `Auto` data set.

(a) Use the `lm()` function to perform a linear regression with `mpg` as the response and `horsepower` as the predictor. Use the `summary()` function to print the results. Comment on the output.

```r
df = Auto
model = lm(data = Auto, mpg~horsepower)
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i) Is there a relationship between the predictor and the response?

There is a relationship between the predictor and the response.

ii) How strong is the relationship between the predictor and the response?

The relationship between the predictor and the response is strong, with a p-value close to 0.

iii) Is the relationship between the predictor and the response positive or negative?

The relationship between the predictor and the response is negative.

iv) What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?

The predicted mpg associated with a horsepower of 98 is

```
new_hp = data.frame(horsepower = 98)
predict.lm(model, new_hp)
```

```
##        1
## 24.46708
```

The associated 95% confidence interval is

```
predict.lm(model, new_hp, interval = "confidence", level = 0.95)
```

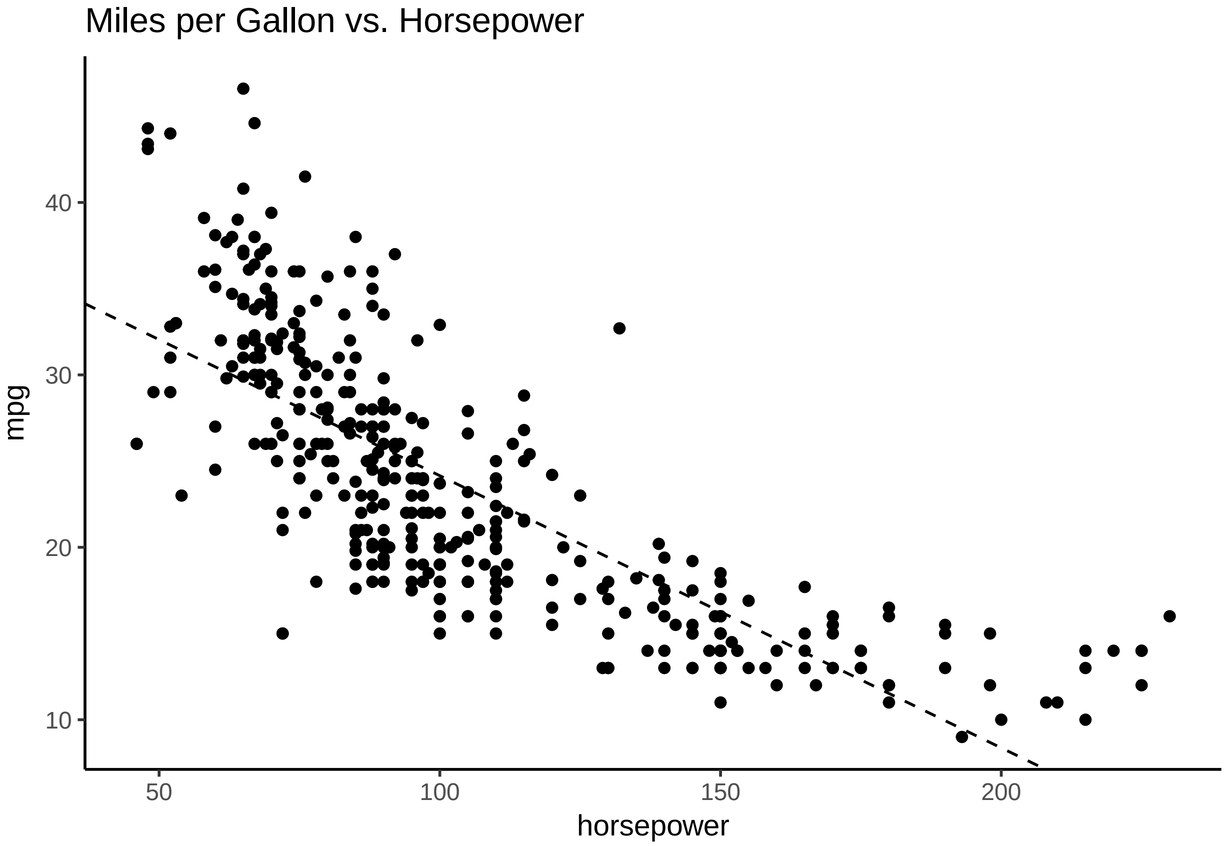
```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```
predict.lm(model, new_hp, interval = "prediction", level = 0.95)
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

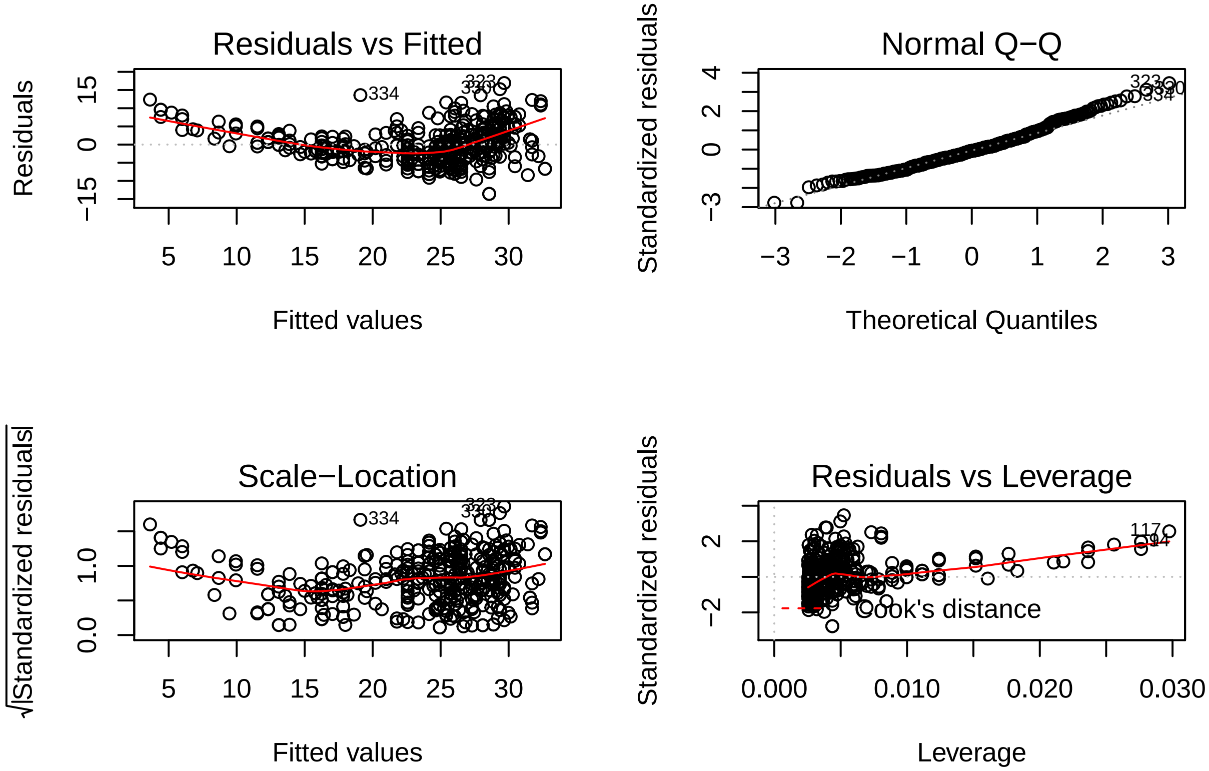
(b) Plot the response and the predictor. Use the abline() function to display the least squares regression line.

```
ggplot(df, aes(x = horsepower, y = mpg)) + geom_point() +
  geom_abline(aes(intercept = model$coefficients[1],
                  slope = model$coefficients[2]),
              linetype = 'dashed') +
  ggtitle("Miles per Gallon vs. Horsepower") +
  theme_classic()
```



(c) Use the `plot()` function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

```
par(mfrow=c(2,2))
plot(model)
```

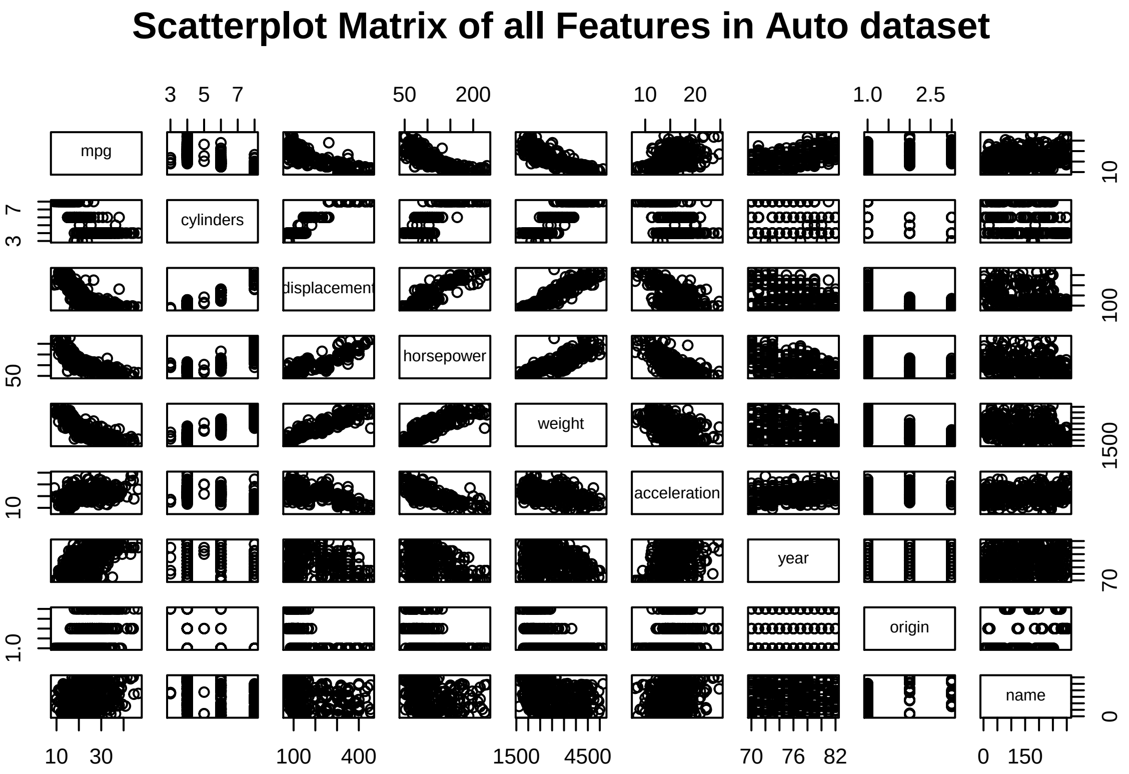


According to the plots above, the residuals are bigger for values that are at the high end as well as low end. This could mean that the relationship is not linear but rather quadratic. In addition, the residuals vs leverage plot shows that most points don't have a huge leverage.

Question 9: This question involves the use of multiple linear regression on the `Auto` data set.

(a) Produce a scatterplot matrix which includes all of the variables in the data set.

```
pairs(df, main = "Scatterplot Matrix of all Features in Auto dataset")
```



(b) Compute the matrix of correlations between the variables using the function `cor()`. You will need to exclude the `name` variable, which is qualitative.

```
cor(select(df, which = -name))
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

(c) Use the `lm()` function to perform a multiple linear regression with `mpg` as the response and all other variables except `name` as the predictors. Use the `summary()` function to print the results. Comment on the output.

```
model = lm(data = df, mpg ~ cylinders + displacement + horsepower +
             weight + acceleration + year + origin)
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i) Is there a relationship between the predictors and the response?

Some predictors have a relationship with the response while some dont.

ii) Which predictors appear to have a statistically significant relationship to the response?

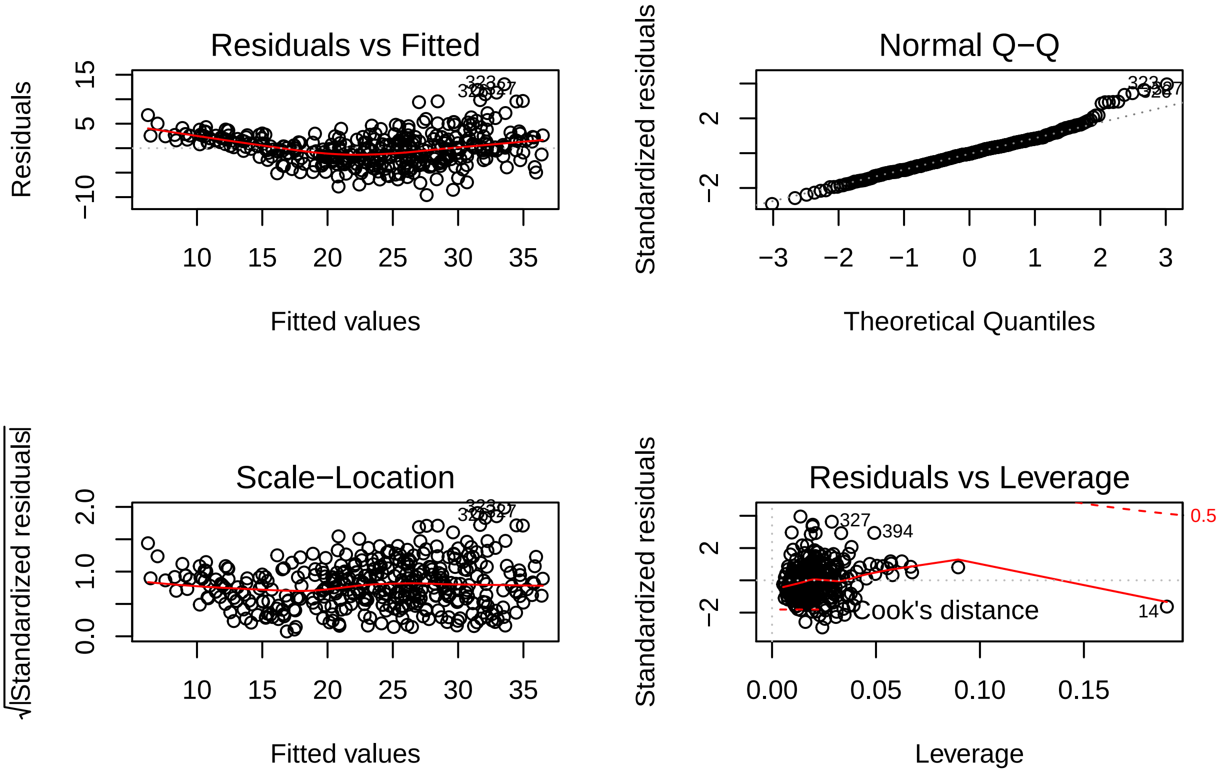
Predictors `displacement`, `weight`, `year` and `origin` are statistically significant. The predictors that have high p-values, such as `cylinders`, `horsepower` and `acceleration` don't show a relationship with `mpg` and so are not statistically significant with the response.

iii) What does the coefficient for the `year` variable suggest?

The coefficient for `year` is 0.75; this suggests that later years have a greater impact on the `mpg` than earlier years.

(d) Use the `plot()` function to produce diagnostic plots of the linear regression fit. Comment on any problems with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusally high leverage?

```r
par(mfrow=c(2,2))
plot(model)
```



The residuals appear to become larger as values become larger. Some residuals appear to very large such as point 323. The leverage plot does identify point 14 with usually high leverage.

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
summary(lm(data = df, mpg~horsepower*acceleration))
```

```
##
## Call:
## lm(formula = mpg ~ horsepower * acceleration, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3442  -2.7324  -0.4049   2.4210  15.8840
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             33.512440   3.420187   9.798  < 2e-16 ***
## horsepower               0.017590   0.027425   0.641 0.521664
## acceleration             0.800296   0.211899   3.777 0.000184 ***
## horsepower:acceleration -0.015698   0.002003  -7.838 4.45e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.426 on 388 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.6784
## F-statistic: 275.9 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
summary(lm(data = df, mpg~displacement*origin))
```

```
##
## Call:
## lm(formula = mpg ~ displacement * origin, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1742  -2.8223  -0.5893   2.2531  18.8420
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         28.41854    1.53883  18.468  < 2e-16 ***
## displacement        -0.01887    0.01082  -1.745  0.08183 .
## origin               4.79247    1.13249   4.232  2.9e-05 ***
## displacement:origin -0.03476    0.01010  -3.442  0.00064 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.526 on 388 degrees of freedom
## Multiple R-squared:  0.6664, Adjusted R-squared:  0.6638
## F-statistic: 258.3 on 3 and 388 DF,  p-value: < 2.2e-16
```

```r
summary(lm(data = df, mpg~cylinders*weight))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * weight, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4916  -2.6225  -0.3927   1.7794  16.7087
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      65.3864559  3.7333137  17.514  < 2e-16 ***
## cylinders        -4.2097950  0.7238315  -5.816 1.26e-08 ***
## weight           -0.0128348  0.0013628  -9.418  < 2e-16 ***
## cylinders:weight  0.0010979  0.0002101   5.226 2.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.165 on 388 degrees of freedom
## Multiple R-squared:  0.7174, Adjusted R-squared:  0.7152
## F-statistic: 328.3 on 3 and 388 DF,  p-value: < 2.2e-16
```

These are some of the interactions that appear to be statistically significant.

(f) Try a few different transformation of the variables, such as $\log(X)$, $\sqrt{X}$, X^2. Comment on the findings.

```r
summary(lm(data = df, mpg~I(log(acceleration))))
```

```
##
## Call:
## lm(formula = mpg ~ I(log(acceleration)), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0234  -5.6231  -0.9787   4.5943  23.0872
```

```
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -27.834      5.373  -5.180 3.56e-07 ***
## I(log(acceleration))   18.801      1.966   9.565  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.033 on 390 degrees of freedom
## Multiple R-squared:   0.19,  Adjusted R-squared:  0.1879 
## F-statistic: 91.49 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
summary(lm(data = df, mpg~poly(cylinders, 3)))
```

```
##
## Call:
## lm(formula = mpg ~ poly(cylinders, 3), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -11.2869  -2.9058  -0.9627   2.3403  18.0218 
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           23.446      0.237  98.919  < 2e-16 ***
## poly(cylinders, 3)1 -120.013      4.693 -25.574  < 2e-16 ***
## poly(cylinders, 3)2    8.113      4.693   1.729   0.0846 .  
## poly(cylinders, 3)3   28.379      4.693   6.047 3.46e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.693 on 388 degrees of freedom
## Multiple R-squared:  0.6413, Adjusted R-squared:  0.6385 
## F-statistic: 231.2 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
summary(lm(data = df, mpg~sqrt(year)))
```

```
##
## Call:
## lm(formula = mpg ~ sqrt(year), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -11.9542  -5.4765  -0.4147   4.9413  18.2363 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -162.712     13.293  -12.24   <2e-16 ***
## sqrt(year)    21.363      1.525   14.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.374 on 390 degrees of freedom
## Multiple R-squared:  0.3347, Adjusted R-squared:  0.333 
## F-statistic: 196.2 on 1 and 390 DF,  p-value: < 2.2e-16
```

These are some of the different transformations that appear to be statistically significant.

Question 10: This question should be answered using the `Carseats` data set.

(a) Fit a multiple regression model to predict `Sales` using `Price`, `Urban` and `US`.

```r
df = Carseats
model = lm(data = df, Sales ~ Price + Urban + US)
model
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = df)
##
## Coefficients:
## (Intercept)        Price     UrbanYes        USYes
##    13.04347     -0.05446     -0.02192      1.20057
```

(b) Provide an interpretation of each coefficient in the model. Be careful - some of the variables in the model are qualitative!

For each dollar increase in price, sales decrease by $54. If the store is in an urban location, then the sales decrease by $21. If the store is in the US, then the sales increase by $1200.

(c) Write out the model in equation form, being careful to handle the qualitative variables properly.

$$\text{Sales} = 13.04347 - 0.05446 * \text{Price} - 0.02192 * \text{UrbanYes} + 1.20057 * \text{USYes}$$

(d) For which of the predictors can you reject the null hypothesis $H_0 : \beta_j = 0$?

```r
summary(model)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

The null hypothesis can be rejected for the `price` and `US` predictors since the p-values are low and so they are statistically significant.

(e) On the basis of the response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

```r
model2 = lm(data = df, Sales~Price + US)
summary(model2)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

(f) How well do the models in (a) and (e) fit the data?

The model in (e) has a slightly lower residual standard error.

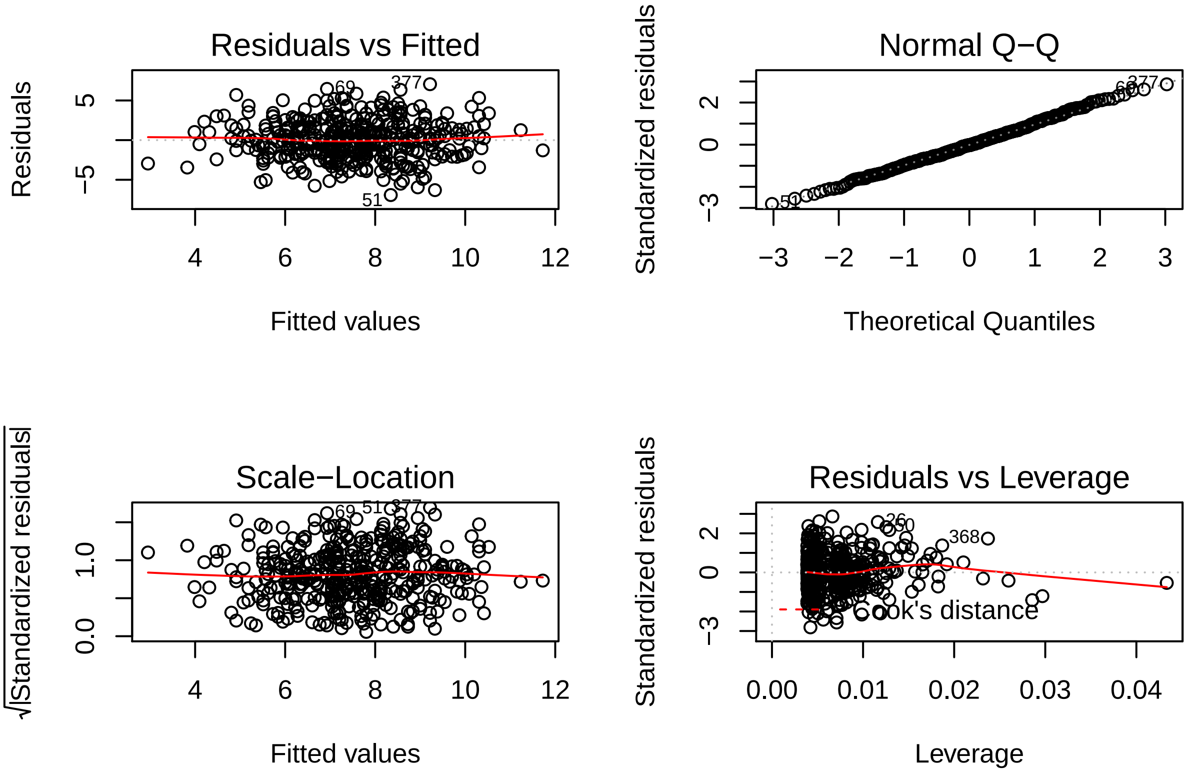
(g) Using the model from (e), obtain 95% confidence intervals for the coefficients(s).

```r
confint(model2)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

(h) Is there evidence of outliers or high leverage observations in the model from (e)?

```r
par(mfrow=c(2,2))
plot(model2)
```



The residuals do not show any abnormal activities, thus there are no outliers. There does appear to be one high leverage point.

Question 11: In this problem, you will investigate the t-statistic for the null hypotheis $H_0 : \beta = 0$ in simple linear regression without an intercept. To begin, generate a predictor x and a response y as follows.

```
set.seed(2019)
x = rnorm(100)
y = 2*x + rnorm(100)
```

(a) Perform a simple linear regression of y onto x, *without* an intercept. Report the coefficient estimate $\hat{\beta}$, the standard error of this coefficient estimate, and the t-statistic and p-value associated with the null hypothesis $H_0 : \beta = 0$. Comment on these results. (To perform regression without an intercept, use the command `lm(y~x+0)`.)

```
summary(lm(y~x+0))
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7810 -0.8142 -0.1381  0.5142  2.1202
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   2.0994     0.1106   18.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9995 on 99 degrees of freedom
## Multiple R-squared:  0.7845, Adjusted R-squared:  0.7823
## F-statistic: 360.4 on 1 and 99 DF,  p-value: < 2.2e-16
```

The coefficent estimate of $\hat{\beta}$ is 2.0994 while the standard error of this estimate is 0.1106. The t-statistic and p-value associated with this coefficent is 18.98 and ≈ 0. This means that the coefficient is statistically significant and that the null hypothesis can be rejected.

(b) Now perform a simple linear regression of `x` onto `y` without an intercept, and report the coefficient estimate, its standard error, and the corresponding t-statistic and p-values associated with the null hypothesis $H_0 : \beta = 0$. Comment on these results.

```r
summary(lm(x~y+0))
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96452 -0.24016 -0.03117  0.25725  1.28889
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.37369    0.01968   18.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4217 on 99 degrees of freedom
## Multiple R-squared:  0.7845, Adjusted R-squared:  0.7823
## F-statistic: 360.4 on 1 and 99 DF,  p-value: < 2.2e-16
```

The coefficent estimate of $\hat{\beta}$ is 0.37369 while the standard error of this estimate is 0.01968. The t-statistic and p-value associated with this coefficent is 18.98 and ≈ 0. This means that the coefficient is statistically significant and that the null hypothesis can be rejected.

(c) What is the relationship between the results obtained in (a) and (b)?

The calculated t-statistic and associated p-value are approximately the same.

(d) For the regression of Y onto X without an intercept, the t-statistic for $H_0 : \beta = 0$ takes the form $\hat{\beta}/\mathrm{SE}[\hat{\beta}]$, where

$$\hat{\beta} = \left(\sum_{i=1}^{n} x_i y_i\right) \Bigg/ \left(\sum_{i'=1}^{n} x_{i'}^2\right)$$

and

$$\mathrm{SE}[\hat{\beta}] = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i\hat{\beta})^2}{(n-1)\sum_{i'=1}^{n} x_{i'}^2}}$$

Confirm numerically that the t-statistic can be written as

$$\frac{(\sqrt{n-1})\sum_{i=1}^{n} x_i y_i}{\sqrt{(\sum_{i=1}^{n} x_i^2)(\sum_{i'=1}^{n} y_{i'}^2) - (\sum_{i'=1}^{n} x_{i'} y_{i'})^2}}$$

```
numerator = sqrt(length(x)-1) * sum(x*y)
denominator = sqrt(sum(x*x) * sum(y*y) - (sum(x*y))^2)
t = numerator / denominator
t
```

```
## [1] 18.98483
```

(e) Using the results from (d), argue that the t-statistic for the regression of `y` onto `x` is the same as the t-statistic for the regression of `x` onto `y`.

When regressing `y` onto `x` or `x` onto `y`, the same correlations are created between the two variables. It then makes sense for the t-statistic to be the same for both scenarios.

(f) Show that when regression is performed *with* an intercept the t-statistic for $H_0 : \beta = 0$ is the same for the regression of `y` onto `x` as it is for the regression of `x` onto `y`.

```
summary(lm(y~x))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59993 -0.65990  0.02888  0.67219  2.29537
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.16401    0.09942   -1.65    0.102
## x            2.08465    0.11000   18.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.991 on 98 degrees of freedom
## Multiple R-squared:  0.7856, Adjusted R-squared:  0.7835
## F-statistic: 359.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(x~y))
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01149 -0.28904 -0.07207  0.21752  1.24390
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04609    0.04260   1.082    0.282
## y            0.37687    0.01989  18.952   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4213 on 98 degrees of freedom
## Multiple R-squared:  0.7856, Adjusted R-squared:  0.7835
## F-statistic: 359.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

The same t-statistics are calculated when the intercept is incorporated for both regressions.

Question 12: This problem involves simple linear regression without an intercept.

(a) Recall that the coefficient estimate $\hat{\beta}$ for the linear regression of Y onto X without an intercept is

$$\hat{\beta} = \left(\sum_{i=1}^{n} x_i y_i\right) / \left(\sum_{i'=1}^{n} x_{i'}^2\right)$$

Under what circumstance is the coefficient estimate for the regression of X onto Y the same as the coefficient estimate for the regression of Y onto X?

The coefficient estimates are the same for both regressions when $\sum x_i^2 = \sum y_i^2$.

(b) Generate an example with $n = 100$ observations in which the coefficient estimate for the regression of X onto Y is *different from* the coefficient estimate for the regression of Y onto X.

```
set.seed(1984)
x = rnorm(100)
y = 2*x + rnorm(100)
lm(y~x)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##    -0.07594      2.09184
```

```
lm(x~y)
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Coefficients:
## (Intercept)            y
##     0.03536      0.38859
```

The coefficient estimates for both regression are not the same.

(c) Generate an example with $n = 100$ observations in which the coefficient estimate for the regression of X onto Y is *the same* as the coefficient estimate for the regression of Y onto X.

```
set.seed(1995)
x = rnorm(100, mean = 50, sd = 1)
y = rnorm(100, mean = 50, sd = 1)
lm(y~x)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     38.2511       0.2375
```

```
lm(x~y)
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Coefficients:
## (Intercept)            y
##     38.2675       0.2323
```

The coefficient estimates for both regressions are nearly the same.

Question 13: In this exercise, you will create some stimulated data and will fit simple linear regression models to it. Set seed to ensure consistent results.

```
set.seed(42)
```

(a) Using the `rnorm()` function, create a vector, `x`, containing 100 observations drawn from a $\mathcal{N}(0,1)$ distribution. This represents a feature, X.

```
x = rnorm(100)
```

(b) Using the `rnorm()` function, create a vector, `eps`, containing 100 observations drawn from a $\mathcal{N}(0,0.25)$ distribution i.e. a normal distribution with mean zero and variance 0.25.

```
eps = rnorm(100, 0, 0.25)
```

(c) Using `x` and `eps`, generate a vector `y` according to the model

$$Y = -1 + 0.5X + \varepsilon$$

```
y = -1 + 0.5*x + eps
```

What is the length of the vector `y`? What are the values of β_0 and β_1 in this linear model?

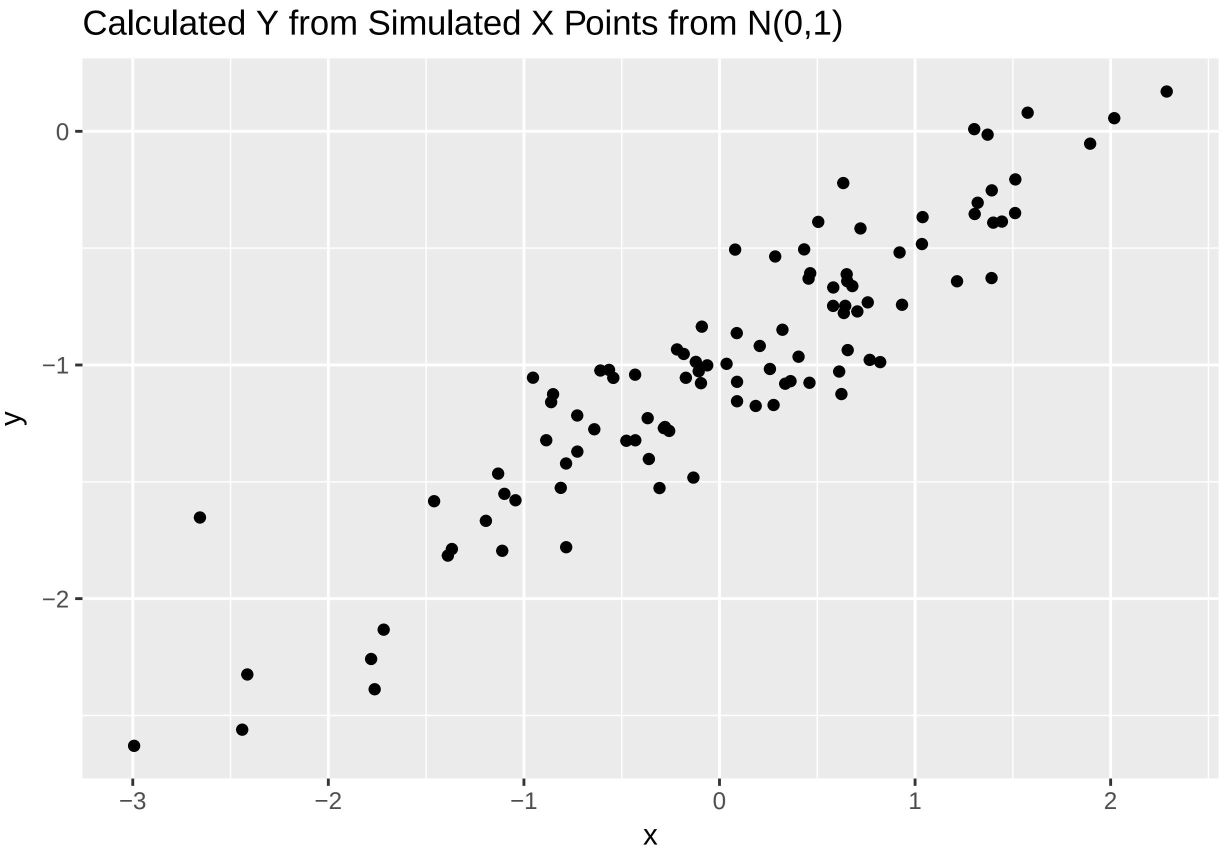
```
length(y)
```

```
## [1] 100
```

The length of the vector `y` is 100. The values of β_0 and β_1 in this linear model are -1 and 0.5 respectively.

(d) Create a scatterplot displaying the relationship between `x` and `y`. Comment some observations.

```
df = data.frame(x,y)
ggplot(data = df, aes(x, y)) +
  geom_point() + ggtitle("Calculated Y from Simulated X Points from N(0,1)")
```



The correlation between X and Y is strong and positive, which makes sense given the equation for Y. There does appear to be one or two outliers.

(e) Fit a least squares linear model to predict `y` using `x`. Comment on the model obtained. How do $\hat{\beta}_0$ and $\hat{\beta}_1$ compare to β_0 and β_1?

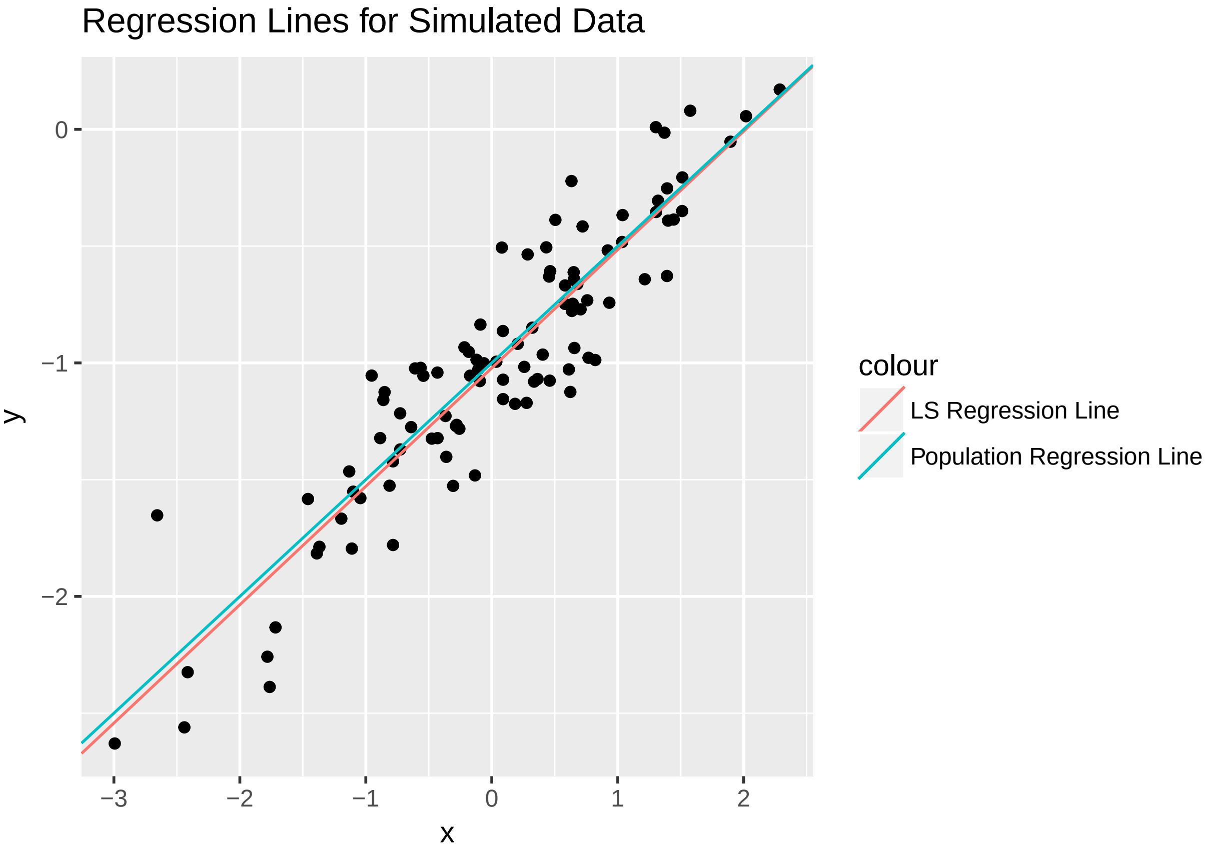
```
model = lm(y~x)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47211 -0.12666  0.00306  0.13527  0.71560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02209    0.02272  -44.99   <2e-16 ***
## x            0.50679    0.02192   23.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2271 on 98 degrees of freedom
## Multiple R-squared:  0.8451, Adjusted R-squared:  0.8435
## F-statistic: 534.7 on 1 and 98 DF,  p-value: < 2.2e-16
```

The model is fairly strong, with an R^2 value of 0.84 and RSE of 0.22. The calculated β values are close to the actual β values. Here $\hat{\beta}_0 = -1.02209$ when $\beta_0 = -1$ and $\hat{\beta}_1 = 0.50679$ when $\beta_1 = 0.5$.

(f) Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the `legend()` command to create an appropriate legend.

```
ggplot(df, aes(x,y)) + geom_point() +
  geom_abline(aes(intercept = model$coefficients[1],
                  slope = model$coefficients[2],
                  col = "LS Regression Line")) +
  geom_abline((aes(intercept = -1,
                   slope = 0.5,
                   col = "Population Regression Line"))) +
  ggtitle("Regression Lines for Simulated Data")
```



(g) Now fit a polynomial regression model that predicts `y` using `x` and `x^2`. Is there evidence that the quadratic term improves the model fit? Explain your answer.

```
model2 = lm(data = df, y~x+poly(x,2))
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x + poly(x, 2), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47440 -0.12491  0.00183  0.13597  0.70686
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02209    0.02283  -44.76   <2e-16 ***
## x             0.50679    0.02203   23.01   <2e-16 ***
## poly(x, 2)1        NA         NA      NA       NA
## poly(x, 2)2  0.02736    0.22823    0.12    0.905
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2282 on 97 degrees of freedom
## Multiple R-squared:  0.8451, Adjusted R-squared:  0.8419
## F-statistic: 264.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

The quadratic term does not appear to improve the model fit, given the low t-statistic and high p-value. This means that the coefficient is statistically insignificant.

(h) Repeat (a)-(f) after modifying the data generative process in such a way that there is *less* noise in the data. The model

$$Y = -1 + 0.5X + \varepsilon$$

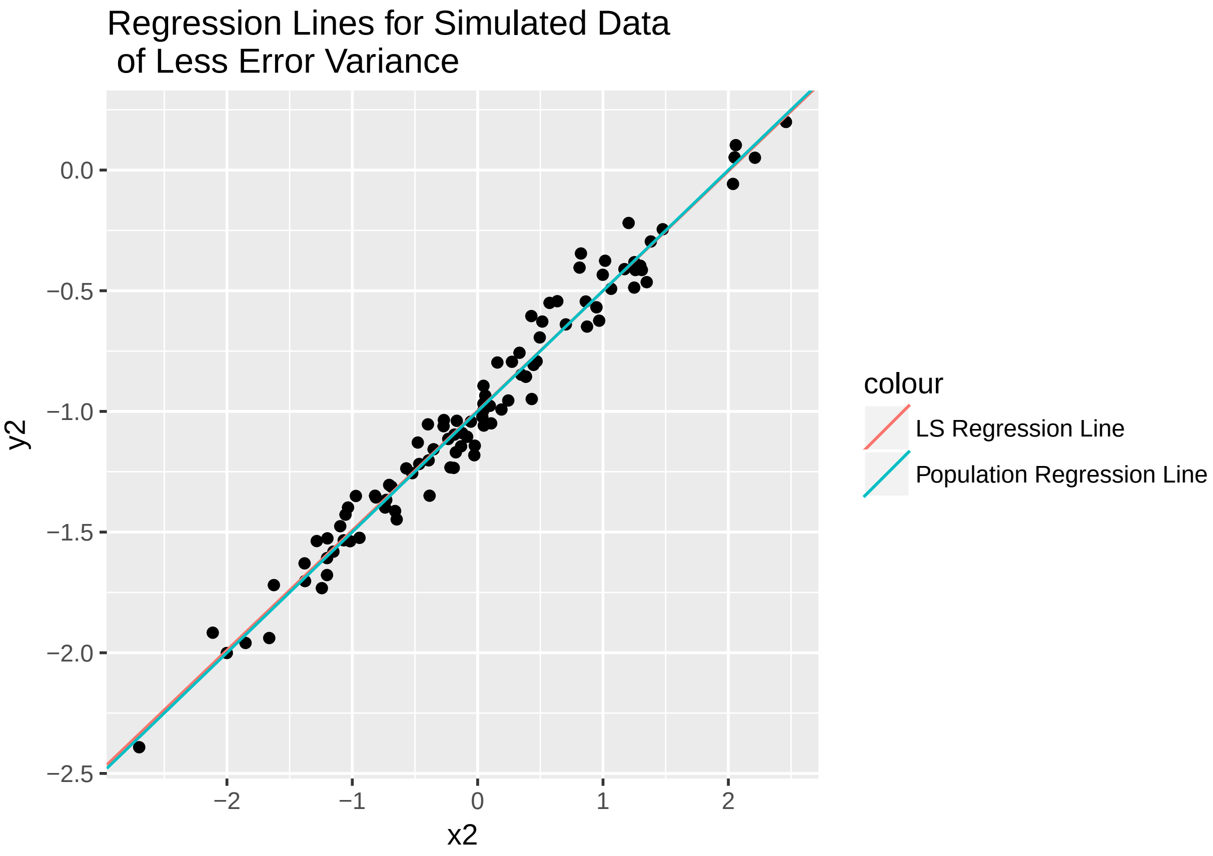
should remain the same. Do this by decreasing the variance of the normal distrbution used to generate the error term ε in (b). Describe the results.

```r
x2 = rnorm(100)
eps2 = rnorm(100, 0, 0.1)
y2 = -1 + 0.5*x2 + eps2

df2 = data.frame(x2, y2)
model2 = lm(data = df2, y2~x2)
summary(model2)
```

```
##
## Call:
## lm(formula = y2 ~ x2, data = df2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.171606 -0.056497 -0.009962  0.064930  0.242228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.996747   0.008798 -113.30   <2e-16 ***
## x2           0.496039   0.008694   57.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08797 on 98 degrees of freedom
## Multiple R-squared:  0.9708, Adjusted R-squared:  0.9705
## F-statistic:  3256 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
ggplot(df2, aes(x2,y2)) + geom_point() +
  geom_abline(aes(intercept = model2$coefficients[1],
                  slope = model2$coefficients[2],
                  col = "LS Regression Line")) +
  geom_abline((aes(intercept = -1,
                   slope = 0.5,
                   col = "Population Regression Line"))) +
  ggtitle("Regression Lines for Simulated Data \n of Less Error Variance")
```



The least squares regression line comes close to the population regression line when there is less variance in the error. The $\hat{\beta}$ are also really close to its respective β values. The residual standard error is lower when there is less error.

(i) Repeat (a)-(f) after modifying the data generative process in such a way that there is *more* noise in the data. The model

$$Y = -1 + 0.5X + \varepsilon$$

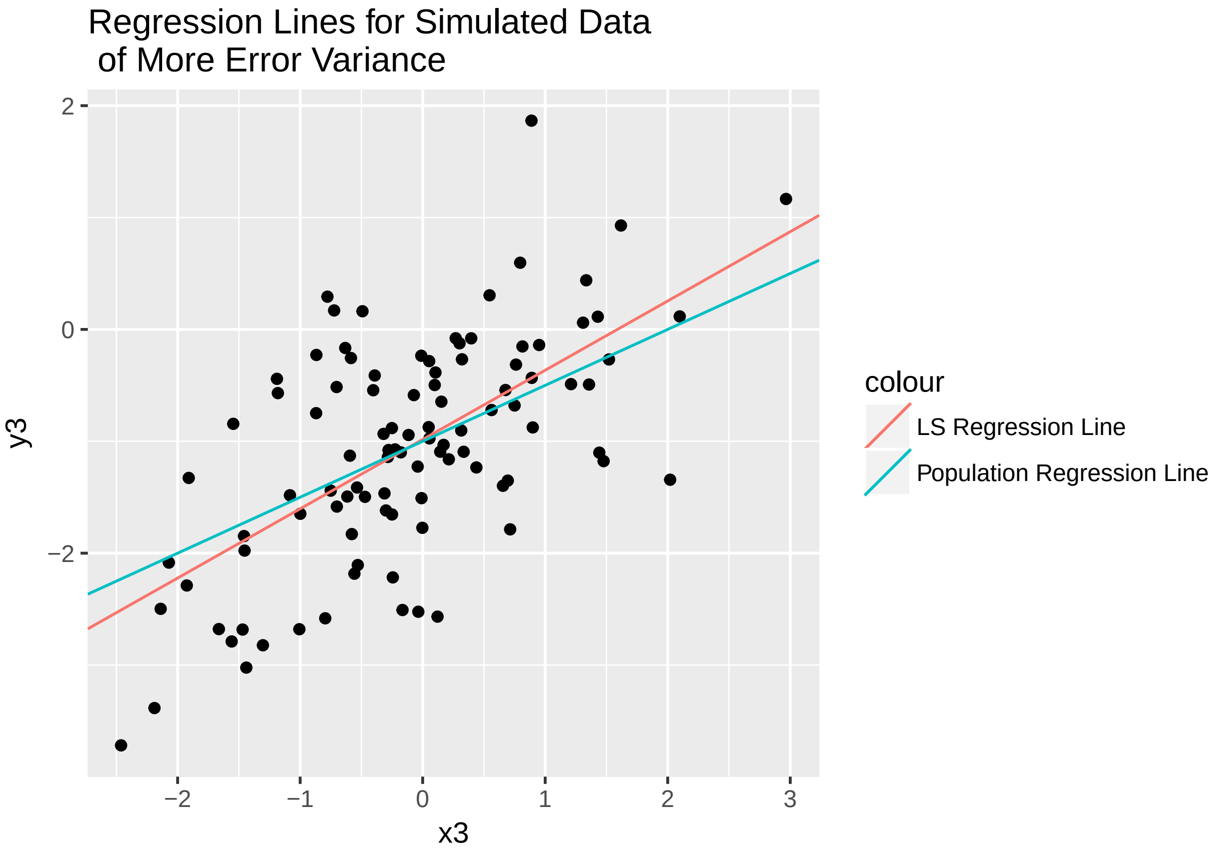
should remain the same. Do this by increasing the variance of the normal distribution used to generate the error term ε in (b). Describe the results.

```
x3 = rnorm(100)
eps3 = rnorm(100, 0, 0.75)
y3 = -1 + 0.5*x3 + eps3

df3 = data.frame(x3, y3)
model3 = lm(data = df3, y3~x3)
summary(model3)
```

```
##
## Call:
## lm(formula = y3 ~ x3, data = df3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65795 -0.49400 -0.01423  0.52228  2.30014
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98419    0.07975 -12.341  < 2e-16 ***
## x3           0.61916    0.07806   7.932 3.54e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7922 on 98 degrees of freedom
## Multiple R-squared:  0.391,  Adjusted R-squared:  0.3848
## F-statistic: 62.92 on 1 and 98 DF,  p-value: 3.539e-12
```

```
ggplot(df3, aes(x3,y3)) + geom_point() +
  geom_abline(aes(intercept = model3$coefficients[1],
                  slope = model3$coefficients[2],
                  col = "LS Regression Line")) +
  geom_abline((aes(intercept = -1,
                   slope = 0.5,
                   col = "Population Regression Line"))) +
  ggtitle("Regression Lines for Simulated Data \n of More Error Variance")
```



The data stimulated here are more sparse in its distribution and does not have a strong linear relationship as above. The estimated $\hat{\beta}$ are also farther from its actual β values than before. The residual standard error is greater than before.

(j) What are the confidence intervals for β_0 and β_1 based on the original data set? the noisier data set, and the less noisy data set? Comment on the results.

```
confint(model)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0671777 -0.9770057
## x             0.4632977  0.5502819
```

```
confint(model2)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0142062 -0.9792886
## x2           0.4787863  0.5132910
```

```
confint(model3)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1424487 -0.8259252
## x3           0.4642568  0.7740588
```

As the data set gets noisier, there is a greater spread in the confidence interval for both coefficients; this means that the parameter was confidently captured in a larger range. On the other hand, as the data set gets less noisier, there is a smaller spread in the confidence interval for both coefficients; this means there is more confidence that the true parameter was captured in a smaller range.

Question 14: This problem focuses on the *collinearity* problem.

(a) Perform the following commands.

```
set.seed(25)
x1 = runif(100)
x2 = 0.5*x1 + rnorm(100)/10
y = 2 + 2*x1 + 0.3*x2 + rnorm(100)
```

The last line corresponds to creating a linear model in which `y` is a function of `x1` and `x2`. Write out the form of the linear model. What are the regression coefficient?

$$y = \beta_0 + 2\beta_1 + 0.3\beta_2 + \varepsilon$$

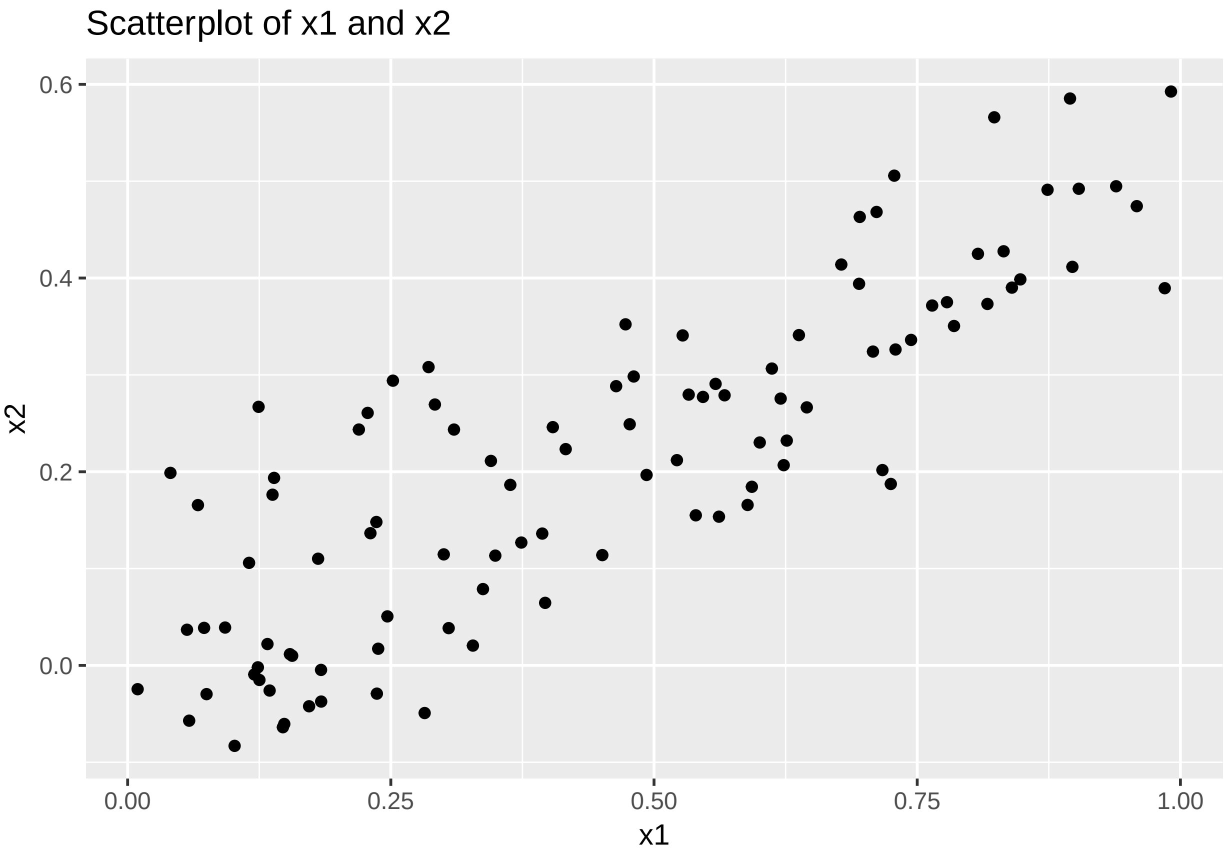
where $\beta_0 = 2$, $\beta_1 = 2$ and $\beta_2 = 0.3$.

(b) What is the correlation between `x1` and `x2`? Create a scatterplot displaying the relationship between the variables.

```
cor(x1,x2)
```

```
## [1] 0.8440404
```

```
df = data.frame(x1, x2, y)
ggplot(data = df, aes(x1, x2)) + geom_point() + ggtitle("Scatterplot of x1 and x2")
```



(c) Using this data, fit a least squares regression to predict y using x1 and x2. Describe the results obtained. What are $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_) : \beta_1 = 0$? How about the null hypothesis $H_0 : \beta_2 = 0$?

```
model = lm(data = df, y~x1 + x2)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62782 -0.69419 -0.06898  0.62083  2.34016
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9157     0.1917   9.993   <2e-16 ***
## x1            1.3653     0.6720   2.032   0.0449 *
## x2            2.5851     1.0865   2.379   0.0193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9917 on 97 degrees of freedom
## Multiple R-squared:  0.3915, Adjusted R-squared:  0.379
## F-statistic: 31.21 on 2 and 97 DF,  p-value: 3.436e-11
```

The model has a RSE of 0.9917 and R^2 value of 0.379. The coefficients found are $\hat{\beta}_0 = 1.9157$, $\hat{\beta}_1 = 1.3653$

and $\hat{\beta}_2 = 2.5851$. The value for β_0 was almost and then coefficient estimates became more and more deviated from the true population coefficients. At the α level 0.01, both null hypotheses cannot be rejected and cannot be said that the coefficient estimates are statistically significant.

(d) Now fit a least squares regression to predict `y` using only `x1`. Comment on the results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
model2 = lm(data = df, y~x1)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x1, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.06463 -0.67723 -0.09991  0.70066  2.64774
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8585     0.1947   9.547 1.16e-15 ***
## x1            2.7149     0.3689   7.360 5.72e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.015 on 98 degrees of freedom
## Multiple R-squared:  0.356,  Adjusted R-squared:  0.3494
## F-statistic: 54.17 on 1 and 98 DF,  p-value: 5.719e-11
```

The model does slightly worse here when removing one of the two predictor variables. The RSE went slightly up while the R^2 value went slightly down. The null hypothesis can be rejected here.

(e) Now fit a least squares regression to predict `y` using only `x2`. Comment on the results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
model3 = lm(data = df, y~x2)
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20520 -0.67113 -0.05196  0.79059  2.29363
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1337     0.1614  13.222  < 2e-16 ***
## x2            4.4483     0.5919   7.515  2.7e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.007 on 98 degrees of freedom
## Multiple R-squared:  0.3656, Adjusted R-squared:  0.3591
## F-statistic: 56.48 on 1 and 98 DF,  p-value: 2.701e-11
```

The same results can be drawn here as above. The model does slightly worse when removing the other predictor variable. The RSE went slightly up while the R^2 value went slightly down. The null hypothesis can also be rejected here.

(f) Do the results obtained in (c)-(e) contradict each other? Explain your answer.

The results in (c)-(e) do contradict each other. When `x1` and `x2` are used together in the linear model, it predicts the model well while showing that the coefficient estimates are not statistically significant at the α level of 0.01. On other hand, when `x1` and `x2` are individually used to create the linear model, it predicts the model poorly but show that the coefficient estimates are statistically significant.

(g) Now suppose there is one additional observation, which was unfortunately mismeasured.

```r
x1 = c(x1, 0.1)
x2 = c(x2, 0.8)
y = c(y, 6)
```

Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on each of the models? In each model, is this observation an outlier? A high leverage point? Both? Explain your answer.

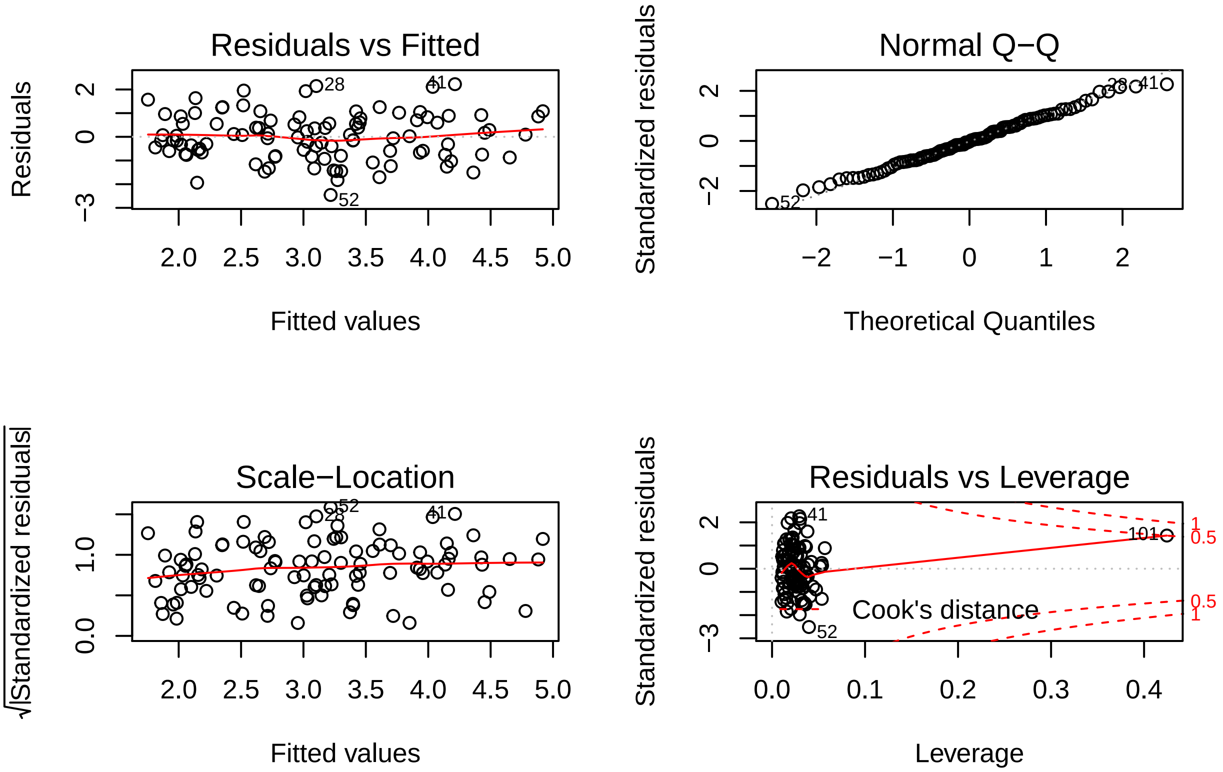
```r
df2 = data.frame(x1, x2, y)

model_new1 = lm(data = df2, y~x1+x2)
model_new2 = lm(data = df2, y~x1)
model_new3 = lm(data = df2, y~x2)

summary(model_new1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45599 -0.73212 -0.02505  0.69838  2.22347
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9711     0.1888  10.440  < 2e-16 ***
## x1            0.7934     0.5446   1.457    0.148
## x2            3.5848     0.8395   4.270 4.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9971 on 98 degrees of freedom
## Multiple R-squared:  0.4103, Adjusted R-squared:  0.3982
## F-statistic: 34.09 on 2 and 98 DF,  p-value: 5.778e-12
```

```r
par(mfrow=c(2,2))
plot(model_new1)
```



```
summary(model_new2)
```

```
##
## Call:
## lm(formula = y ~ x1, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0544 -0.6883 -0.0933  0.6919  3.7711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9748     0.2046   9.653 6.26e-16 ***
## x1            2.5404     0.3895   6.522 2.95e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.08 on 99 degrees of freedom
## Multiple R-squared:  0.3005, Adjusted R-squared:  0.2935
## F-statistic: 42.54 on 1 and 99 DF,  p-value: 2.952e-09
```

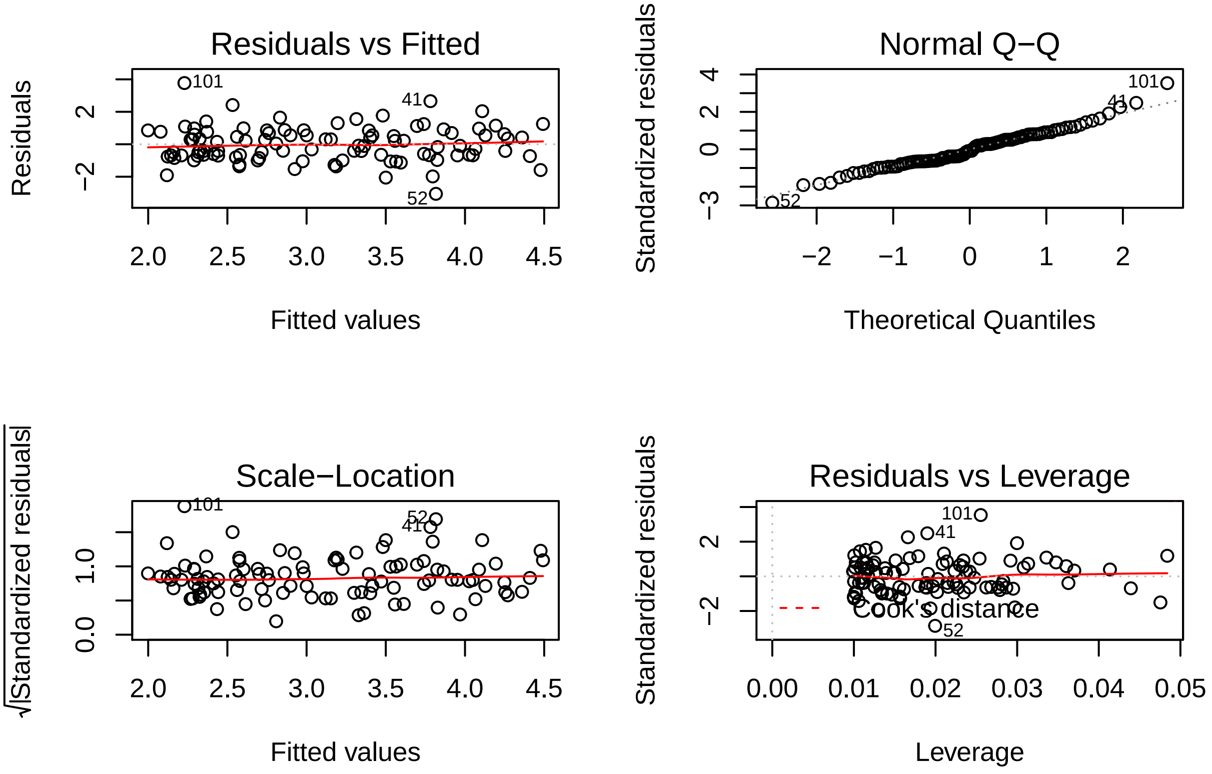
```
par(mfrow=c(2,2))
plot(model_new2)

summary(model_new3)
```

```
##
## Call:
## lm(formula = y ~ x2, data = df2)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20651 -0.67070 -0.04967  0.78643  2.29248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1247     0.1575  13.486  < 2e-16 ***
## x2            4.5035     0.5572   8.082  1.6e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.003 on 99 degrees of freedom
## Multiple R-squared:  0.3975, Adjusted R-squared:  0.3914
## F-statistic: 65.32 on 1 and 99 DF,  p-value: 1.602e-12
```

```
par(mfrow=c(2,2))
plot(model_new2)
```



When using both `x1` and `x2`, the R^2 value does go up when incorporating the new value. Observation 101 is a high leverage point. When using only `x1` or `x2`, observation 101 is a high leverage point.

Question 15: This problem involves the `Boston` data set. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response and the other variables are the predictors.

(a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there is a statistically significant association between the predictor and the predictor?

```r
df = Boston
model1 = lm(data = df, crim~zn)
model2 = lm(data = df, crim~indus)
model3 = lm(data = df, crim~chas)
model4 = lm(data = df, crim~nox)
model5 = lm(data = df, crim~rm)
model6 = lm(data = df, crim~age)
model7 = lm(data = df, crim~dis)
model8 = lm(data = df, crim~rad)
model9 = lm(data = df, crim~tax)
model10 = lm(data = df, crim~ptratio)
model11 = lm(data = df, crim~black)
model12 = lm(data = df, crim~lstat)
model13 = lm(data = df, crim~medv)

summary(model1)$coefficients[2,4]
```

```
## [1] 5.506472e-06
```

```r
summary(model2)$coefficients[2,4]
```

```
## [1] 1.450349e-21
```

```r
summary(model3)$coefficients[2,4]
```

```
## [1] 0.2094345
```

```r
summary(model4)$coefficients[2,4]
```

```
## [1] 3.751739e-23
```

```r
summary(model5)$coefficients[2,4]
```

```
## [1] 6.346703e-07
```

```r
summary(model6)$coefficients[2,4]
```

```
## [1] 2.854869e-16
```

```r
summary(model7)$coefficients[2,4]
```

```
## [1] 8.519949e-19
```

```r
summary(model8)$coefficients[2,4]
```

```
## [1] 2.693844e-56
```

```r
summary(model9)$coefficients[2,4]
```

```
## [1] 2.357127e-47
```

```r
summary(model10)$coefficients[2,4]
```

```
## [1] 2.942922e-11
```

```r
summary(model11)$coefficients[2,4]
```

```
## [1] 2.487274e-19
```

```r
summary(model12)$coefficients[2,4]
```

```
## [1] 2.654277e-27
```

```
summary(model13)$coefficients[2,4]
```

```
## [1] 1.173987e-19
```

All of the coefficient estimates are statistically significant at the α level of 0.01 except `chas`.

(b) 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 $H_0 : \beta_j = 0$?

```
model = lm(data = df, crim~.)
summary(model)
```

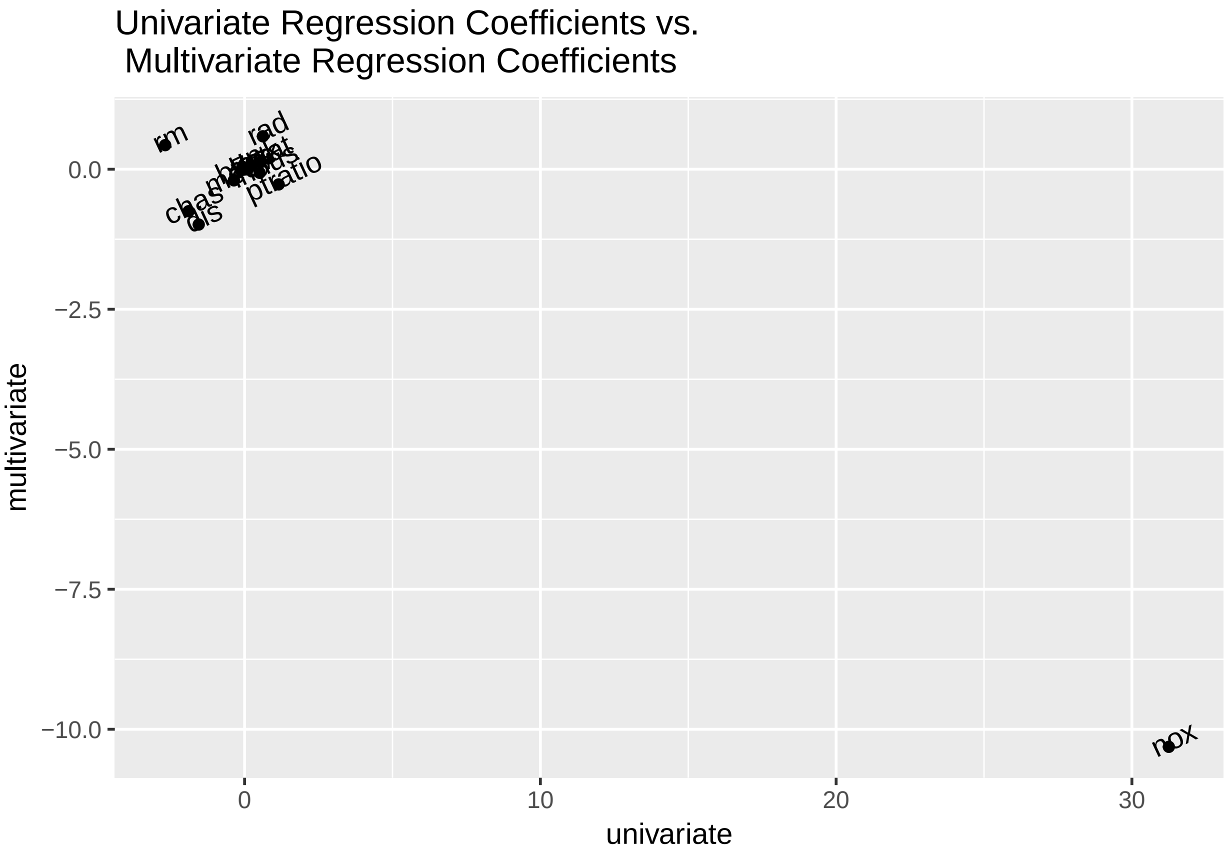
```
##
## Call:
## lm(formula = crim ~ ., data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

The model has a RSE value of 6.439 and R^2 value of 0.454. The coefficient estimates that are statistically significant are `dis`, `rad`, and `medv` and thus their null hypotheses can be rejected.

(c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

```
univariate_coeffs = c(summary(model1)$coefficients[2,1], summary(model2)$coefficients[2,1],
                      summary(model3)$coefficients[2,1], summary(model4)$coefficients[2,1],
                      summary(model5)$coefficients[2,1], summary(model6)$coefficients[2,1],
                      summary(model7)$coefficients[2,1], summary(model8)$coefficients[2,1],
                      summary(model9)$coefficients[2,1], summary(model10)$coefficients[2,1],
                      summary(model11)$coefficients[2,1], summary(model12)$coefficients[2,1],
                      summary(model13)$coefficients[2,1])
multivariate_coeffs = summary(model)$coefficients[2:14,1]
coeffs = data.frame(univariate = univariate_coeffs, multivariate = multivariate_coeffs)

ggplot(data = coeffs, aes(x = univariate, y = multivariate, label = rownames(coeffs))) + geom_point() +
  ggtitle("Univariate Regression Coefficients vs. \n Multivariate Regression Coefficients") +
  geom_text(vjust = 0.5, nudge_x = 0.15, nudge_y = 0.15, angle = 25) #+ xlim(-3,2) + ylim(-1.5, 1)
```

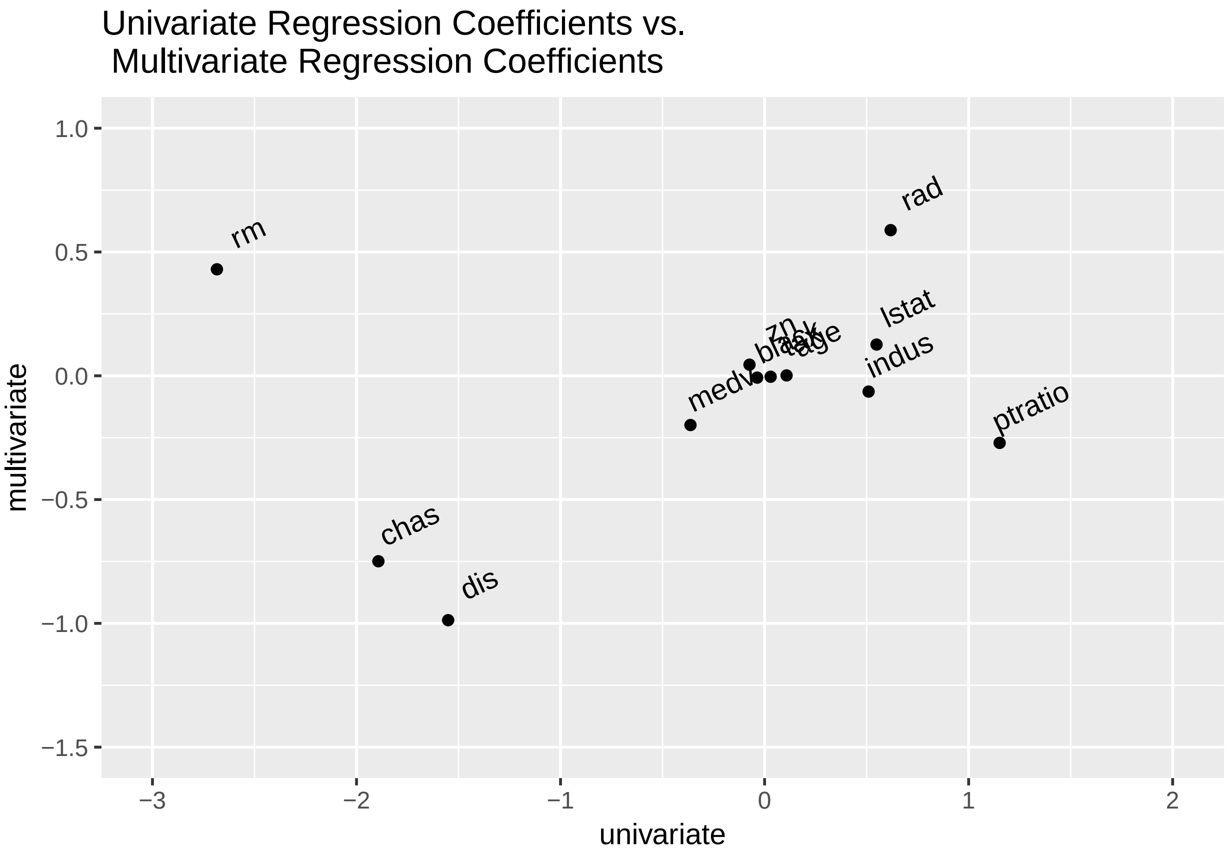


The predictor `nox` had a vastly different coefficient estimate between the univariate regression and the multivariate regression. Omit this point to look at the other predictors in a closer view.

```
ggplot(data = coeffs, aes(x = univariate, y = multivariate, label = rownames(coeffs))) + geom_point() +
  ggtitle("Univariate Regression Coefficients vs. \n Multivariate Regression Coefficients") +
  geom_text(vjust = 0.5, nudge_x = 0.15, nudge_y = 0.15, angle = 25) + xlim(-3,2) + ylim(-1.5, 1)
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```



(d) Is there evidence of 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

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

```
summary(lm(data = df, crim~poly(zn,3)))$coefficients[,4]
```

```
##  (Intercept) poly(zn, 3)1 poly(zn, 3)2 poly(zn, 3)3
## 1.547150e-20 4.697806e-06 4.420507e-03 2.295386e-01
```

```
summary(lm(data = df, crim~poly(indus,3)))$coefficients[,4]
```

```
##     (Intercept) poly(indus, 3)1 poly(indus, 3)2 poly(indus, 3)3
##    3.606468e-25    8.854243e-24    1.086057e-03    1.196405e-12
```

```
#summary(lm(data = df, crim~poly(chas,3)))$coefficients[,4]
summary(lm(data = df, crim~poly(nox,3)))$coefficients[,4]
```

```
##   (Intercept) poly(nox, 3)1 poly(nox, 3)2 poly(nox, 3)3
##  2.742908e-26  2.457491e-26  7.736755e-05  6.961110e-16
```

```
summary(lm(data = df, crim~poly(rm,3)))$coefficients[,4]
```

```
##  (Intercept) poly(rm, 3)1 poly(rm, 3)2 poly(rm, 3)3
## 1.026665e-20 5.128048e-07 1.508545e-03 5.085751e-01
```

```
summary(lm(data = df, crim~poly(age,3)))$coefficients[,4]
```

```
##   (Intercept) poly(age, 3)1 poly(age, 3)2 poly(age, 3)3
##  5.918933e-23  4.878803e-17  2.291156e-06  6.679915e-03
```

```
summary(lm(data = df, crim~poly(dis,3)))$coefficients[,4]
```

```
##   (Intercept) poly(dis, 3)1 poly(dis, 3)2 poly(dis, 3)3 
##  1.060226e-25  1.253249e-21  7.869767e-14  1.088832e-08
```

```
summary(lm(data = df, crim~poly(rad)))$coefficients[,4]
```

```
##  (Intercept)    poly(rad) 
## 9.143174e-30 2.693844e-56
```

```
summary(lm(data = df, crim~poly(tax,3)))$coefficients[,4]
```

```
##   (Intercept) poly(tax, 3)1 poly(tax, 3)2 poly(tax, 3)3 
##  8.955923e-29  6.976314e-49  3.665348e-06  2.438507e-01
```

```
summary(lm(data = df, crim~poly(ptratio,3)))$coefficients[,4]
```

```
##       (Intercept) poly(ptratio, 3)1 poly(ptratio, 3)2 poly(ptratio, 3)3 
##      1.270767e-21      1.565484e-11      2.405468e-03      6.300514e-03
```

```
summary(lm(data = df, crim~poly(black,3)))$coefficients[,4]
```

```
##     (Intercept) poly(black, 3)1 poly(black, 3)2 poly(black, 3)3 
##    2.139710e-22    2.730082e-19    4.566044e-01    5.436172e-01
```

```
summary(lm(data = df, crim~poly(lstat,3)))$coefficients[,4]
```

```
##     (Intercept) poly(lstat, 3)1 poly(lstat, 3)2 poly(lstat, 3)3 
##    4.939398e-24    1.678072e-27    3.780418e-02    1.298906e-01
```

```
summary(lm(data = df, crim~poly(medv,3)))$coefficients[,4]
```

```
##    (Intercept) poly(medv, 3)1 poly(medv, 3)2 poly(medv, 3)3 
##   7.024110e-31   4.930818e-27   2.928577e-35   1.046510e-12
```

```
colnames(df)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"    
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

There is non-linear association between the response and

- `zn`, of degree 1 and 2
- `indus`, of degree 1, 2 and 3
- `nox`, of degree 1, 2 and 3
- `rm`, of degree 1 and 2
- `age`, of degree 1, 2 and 3
- `dis`, of degree 1, 2 and 3
- `rad`, of degree 1
- `tax`, of degree 1 and 2
- `ptratio`, of degree 1, 2 and 3
- `black`, of degree 1
- `lstat`, of degree 1 and 2
- `medv`, of degree 1, 2 and 3

Note: `chas` cannot be put into polynomial form since it is a factor.

All of the practice applied exercises in this document are taken from “An Introduction to Statistical Learning, with applications in R” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.