Ch.3 Exercises: Linear Regression

Conceptual

1. The Null hypothesis for Table 3.4 is :
$$(TV)H_0 : \beta_0 = 0$$
$$(Radio)H_0 : \beta_1 = 0$$
$$(Newspaper)H_0 : \beta_2 = 0$$
The p-values for TV and Radio are significant, so their null hypothesis can be rejected. It shows a strong likelihood that TV and Radio advert spending impacts sales positively. The p-value for Newspaper is not significant and so the null hypothesis is accepted. This shows that newspaper spending does not increase sales in the presence of TV and Radio spending.

2. KNN refers to a non-parametric method that can be used for classification or regression.

   In the case of classification, a KNN classifier identifies the nearest $\boldsymbol{K}$ points to the observation x_0 to be classified. It then estimates the probability of x_0 belonging to a specific class based on the fraction of that class amongst all the selected $\boldsymbol{K}$ points. Finally, the point is classified as belonging to the class with the highest probability. The KNN classifier provided a qualitative response.

   KNN regression is similar to classification in that to also uses the nearest $\boldsymbol{K}$ points to x_0.However, unlike a classifier it provides a quantitative prediction by averaging the $\boldsymbol{K}$ points to estimate the $f(x_0)$.

3. The regression formula for the response and predictors is : `Y = 50 + 20*GPA + 0.07*IQ + 35*Gender + 0.01*GPA:IQ - 10*GPA:Gender`. We can then calculate income for both genders using various predictors.

   (a) iii is True; As males earn more on average than females after their GPA exceeds 3.5.

   (b) 137.1K.

   (c) False; In the case of the female with an IQ of 100 and a GPA of 4.0, the interaction term adds 17.6k to her final salary, and this represents around 15% of her final salary, therefore the impact of the interaction term is substantial but we have calculate the p-value of the coefficient to determine if it is statistically significant.

4. (a) The extra polynomial terms allow for a closer fit (more degrees of freedom) of the training data, so I would expect the training RSS for cubic regression to be lower than for simple linear regression.

   (b) The true relationship is linear and so simple linear regression would generalize better to unseen data, as such I would expect it to have lower test RSS. The cubic model likely over fit the training data, and so I would expect it to have a higher test RSS.

   (c) Cubic regression will have a better fit to non-linear data and so its training RSS will be lower.

   (d) The test RSS depends on how far from linear the true relationship$f(x)$ is. If $f(x)$ is more linear than cubic, then cubic regression can over fit, so cubic RSS will be higher and liner RSS will be lower.If $f(x)$ is more cubic than linear, then linear regression can under fit, so linear RSS will be higher and cubic RSS will be lower.

5.
$$\hat{y}_i = x_i \times \frac{\sum_{i'=1}^{n}(x_{i'}y_{i'})}{\sum_{j=1}^{n} x_j^2}$$

$$\hat{y}_i = \sum_{i'=1}^{n} \frac{(x_{i'}y_{i'}) \times x_i}{\sum_{j=1}^{n} x_j^2}$$

$$\hat{y}_i = \sum_{i'=1}^{n} \left(\frac{x_i x_{i'}}{\sum_{j=1}^{n} x_j^2} \times y_{i'} \right)$$

$$a_{i'} = \frac{x_i x_{i'}}{\sum_{j=1}^{n} x_j^2}$$

6. If $x_i = \bar{x}$, then $\hat{\beta}_1 = 0$ (Eqn 3.4, pg.62). Therefore, $\hat{\beta}_0 = \bar{y}$ and $\hat{y}_i = \bar{y}$. A line will always pass through $(\bar{x}, \bar{y})$ when $x_i = \bar{x}$.

7. When $(\bar{x} = \bar{y} = 0)$ then:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
$$TSS = \sum_{i=1}^{n} y_i^2$$

Our aim is to show that $R^2 = Cor(X,Y)^2$

$$R^2 = 1 - RSS/TSS$$
$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i)^2}$$
$$\hat{y}_i = \hat{\beta}_1 x_i$$

After replacing $\hat{y}_i$ and simplifying:

$$R^2 = 1 - \frac{\sum(\ y_i - (\sum\ x_i y_i / \sum\ x_i^2) x_i)^2}{\sum(y_i)^2} = \frac{2(\sum\ x_i y_i)^2 / \sum\ x_i^2 - \sum(\ x_i y_i)^2 / \sum\ x_i^2}{\sum\ y_i^2}$$

Finally:

$$R^2 = \frac{\sum_{i=1}^{n}\ (x_i y_i)^2}{\sum_{i=1}^{n}\ (x_i)^2 \sum_{i=1}^{n}\ (y_i)^2} = Cor(X,Y)^2$$

Applied

8. (a)

```r
library(ISLR)
data(Auto)
mpg_pwr = lm(mpg~horsepower,data=Auto)
summary(mpg_pwr)
```

```
##
## 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) There is strong evidence of a relationship between mpg and horsepower as the p-value for horsepower's coefficient is close to zero.

(ii) The R^2 statistic is 0.61, and this means 60% of variance in mpg can be explained by horsepower in this model. To calculate the residual error relative to the response we use the mean of mpg and the RSE. The mean of mpg is `23.4459184`.The RSE was 4.906 which indicates a percentage error of`20.9247508`%. The relationship between mpg and horsepower is reasonably strong.

(iii) MPG has a negative linear relationship with horsepower. For every unit increase in horsepower, the mpg falls by -0.158mpg.

(iv)

```r
predict(mpg_pwr, data.frame(horsepower=c(98)), interval='prediction')
```

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

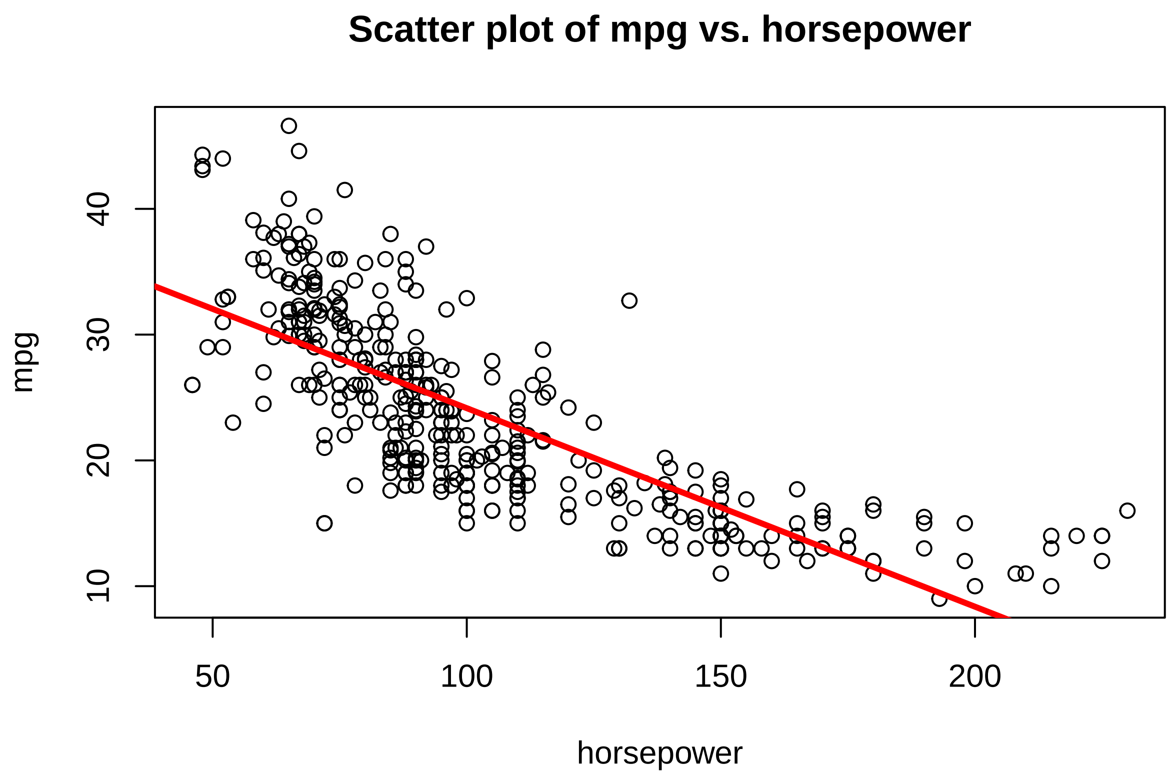
```r
predict(mpg_pwr, data.frame(horsepower=c(98)), interval='confidence')
```

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

- `y = 39.935 - (0.158 x 98) = 24.45`mpg.

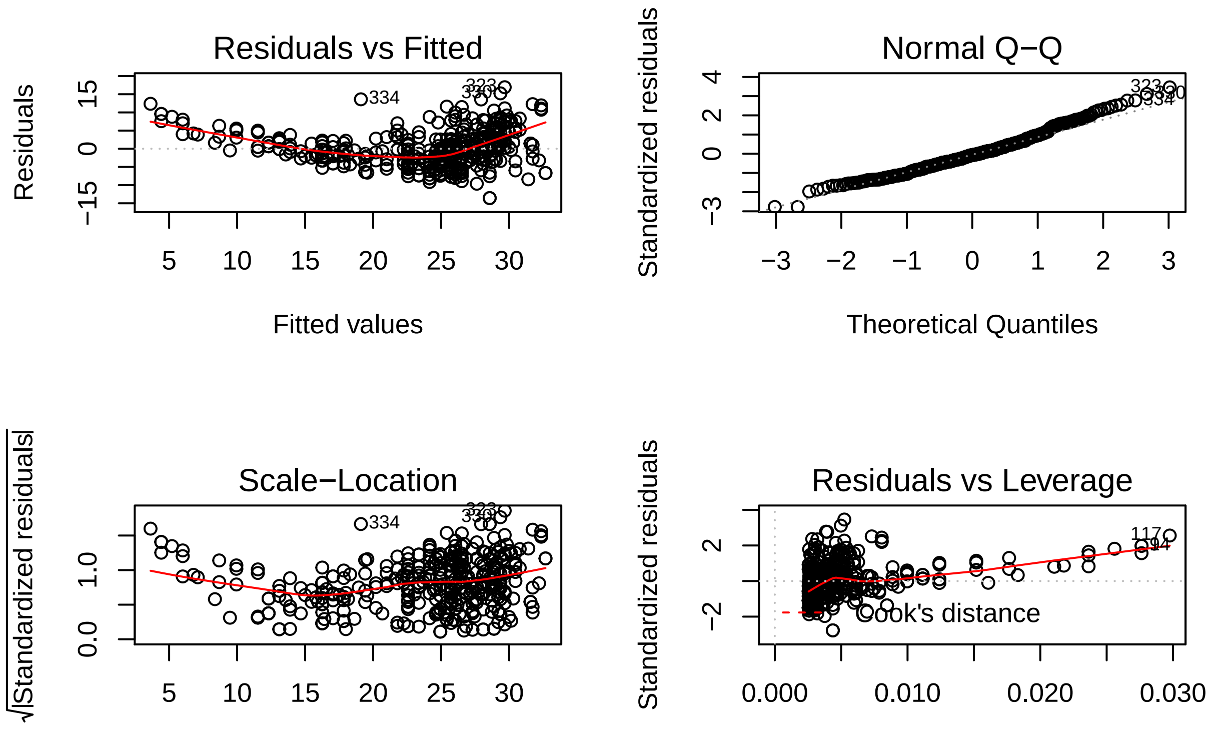
(b)

```r
plot(mpg~horsepower,main= "Scatter plot of mpg vs. horsepower", data=Auto)
abline(mpg_pwr, lwd =3, col ="red")
```



(c)

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



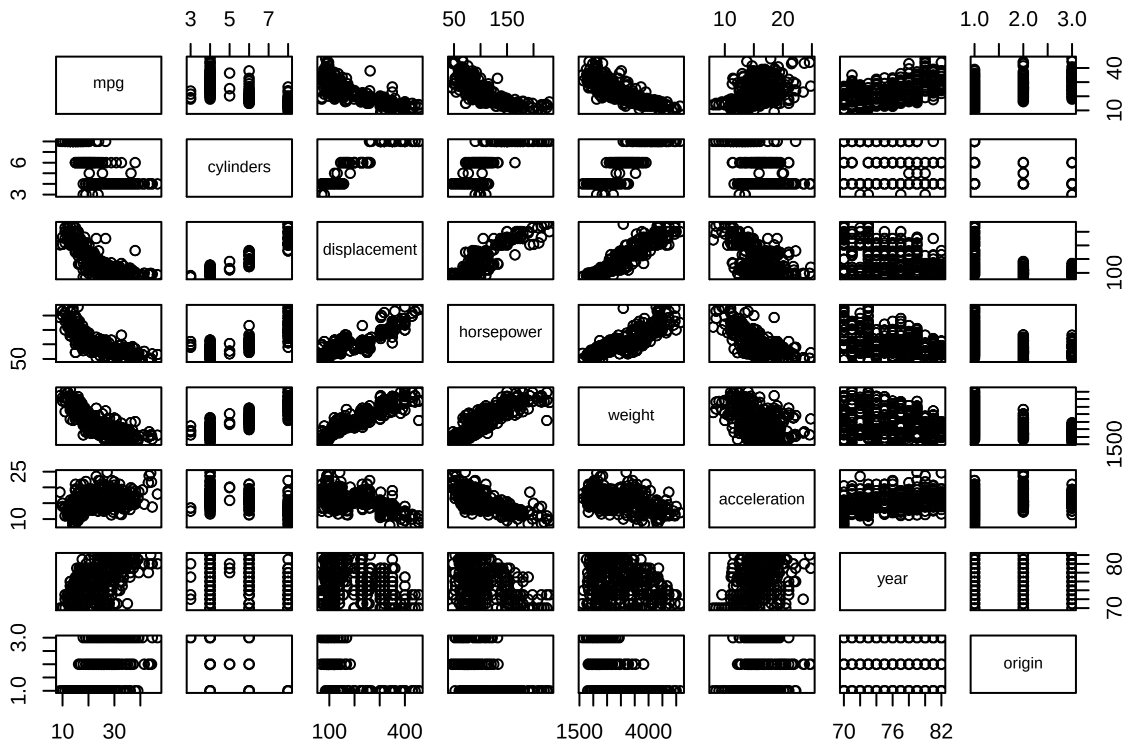
- The residuals v fitted chart shows a slight u-shaped pattern, and this indicates non-linearity in the data.
- The residuals v leverage chart shows that some observations have high leverage.
- The scale-location chart shows some possible outliers. We can confirm by using studentized residuals to find observations with values greater than 3:

```
rstudent(mpg_pwr)[which(rstudent(mpg_pwr)>3)]
```

```
##      323      330
## 3.508709 3.149671
```

9. (a)

```
#head(Auto)
pairs(Auto[,1:8])
```



(b)

```r
cor(Auto[,1:8])
```

```
##                     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)

```r
mpg_all = lm(mpg~.-name,data=Auto)
summary(mpg_all)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## 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
```

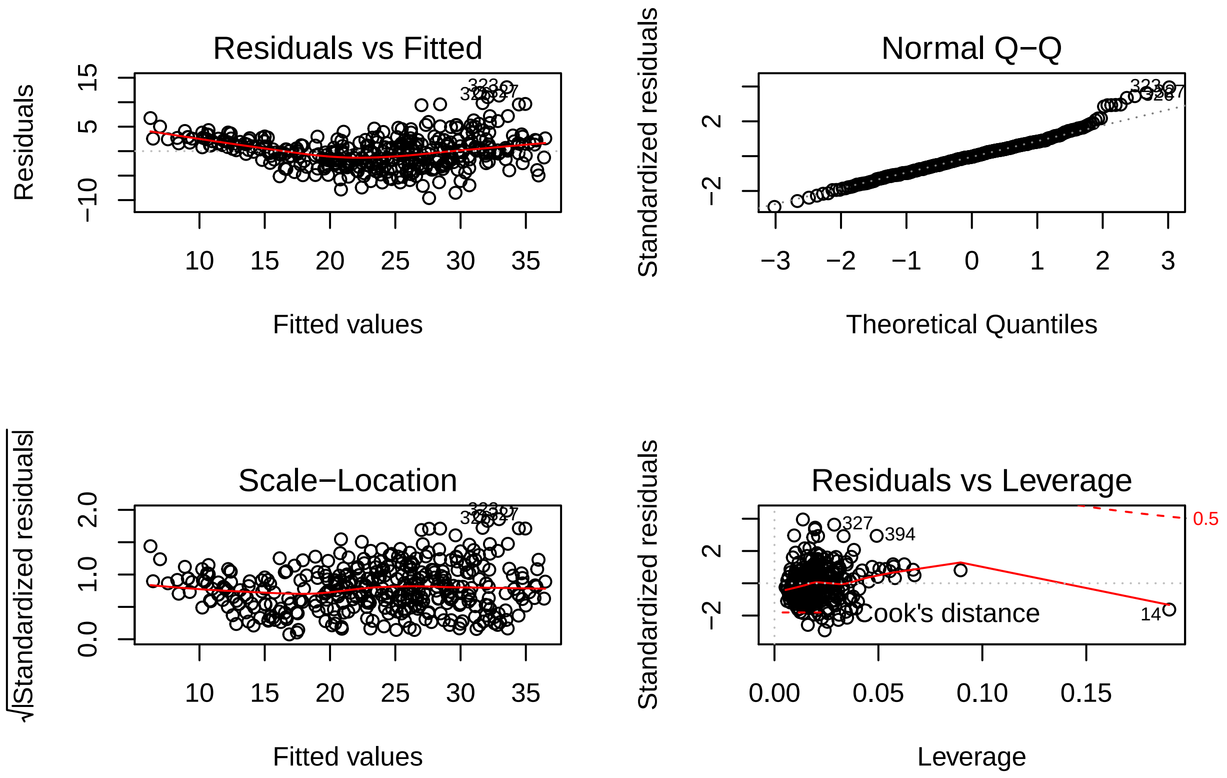
i. p-value: $<$ 3.3e-16 for a F-statistic of 252.4. This shows significant evidence of a relationship between the predictors and the mpg.

ii. Displacement, weight, year and origin are statistically significant as their p-values are below 0.05 or near zero.

iii. The coefficient for the 'year' predictor is 0.75, and suggests that increasing it by one year will mean a vehicles predicted mpg will be 0.75mpg higher, assuming all other predictors are kept constant.

(d)

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



- The Residuals v Fitted values chart indicates some non-linearity in the data, and so polynomial regression could provide a better fit than simple linear regression.
- The standardized residuals vs leverage chart shows that observation 14 has high leverage.
- Some observations are potential outliers:

```
rstudent(mpg_all)[which(rstudent(mpg_all)>3)]
```

```
##      245      323      326      327 
## 3.390068 4.029537 3.494823 3.690246
```

- Additionally, we can check for collinearity between the predictors by finding `VIF` values. The results show some variables with VIF higher than 5 or 10, which according to the text is problematic. Collinearity increases inaccuracy in the estimate for predictors coefficients, and hence increases their standard errors.

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.2
```

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

```r
vif(mpg_all)
```

```
##    cylinders displacement   horsepower       weight acceleration         year 
##    10.737535    21.836792     9.943693    10.831260     2.625806     1.244952 
##       origin 
##     1.772386
```

(e)

```r
mpg_interaction = lm(mpg~.-name + year:cylinders + acceleration:horsepower,data=Auto)
summary(mpg_interaction)
```

```
## 
## Call:
## lm(formula = mpg ~ . - name + year:cylinders + acceleration:horsepower, 
##     data = Auto)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -8.0203 -1.7318 -0.1015  1.5639 11.9559 
## 
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             -8.862e+01  1.212e+01  -7.311 1.57e-12 ***
## cylinders                1.181e+01  2.349e+00   5.029 7.61e-07 ***
## displacement            -8.775e-03  7.916e-03  -1.108  0.26838    
## horsepower               9.151e-02  2.502e-02   3.658  0.00029 ***
## weight                  -4.269e-03  6.967e-04  -6.127 2.23e-09 ***
## acceleration             8.439e-01  1.590e-01   5.306 1.90e-07 ***
## year                     1.521e+00  1.590e-01   9.569  < 2e-16 ***
## origin                   1.070e+00  2.609e-01   4.102 5.00e-05 ***
## cylinders:year          -1.520e-01  3.017e-02  -5.037 7.32e-07 ***
## horsepower:acceleration -9.837e-03  1.778e-03  -5.533 5.84e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.049 on 382 degrees of freedom
## Multiple R-squared:  0.8509, Adjusted R-squared:  0.8474 
## F-statistic: 242.2 on 9 and 382 DF,  p-value: < 2.2e-16
```

- `cylinders:year` and `horsepower:acceleration` are statistically significant. The R^2 metric has increased from 0.82 to 0.85.

```r
mpg_poly = lm(mpg~.-name + year:cylinders + I(horsepower^2)
+ I(acceleration^2),data=Auto)
summary(mpg_poly)
```

```
## 
## Call:
## lm(formula = mpg ~ . - name + year:cylinders + I(horsepower^2) + 
##     I(acceleration^2), data = Auto)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9986 -1.5525 -0.1194  1.4348 11.7722
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.394e+01  1.430e+01  -2.374 0.018075 *
## cylinders          8.481e+00  2.340e+00   3.624 0.000329 ***
## displacement      -1.106e-02  7.330e-03  -1.509 0.132051
## horsepower        -2.720e-01  3.531e-02  -7.703 1.16e-13 ***
## weight            -3.338e-03  6.812e-04  -4.900 1.42e-06 ***
## acceleration      -1.378e+00  5.421e-01  -2.542 0.011403 *
## year               1.272e+00  1.594e-01   7.982 1.71e-14 ***
## origin             1.027e+00  2.493e-01   4.121 4.63e-05 ***
## I(horsepower^2)    8.040e-04  1.140e-04   7.054 8.22e-12 ***
## I(acceleration^2)  3.351e-02  1.578e-02   2.124 0.034303 *
## cylinders:year    -1.056e-01  3.023e-02  -3.493 0.000533 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.935 on 381 degrees of freedom
## Multiple R-squared:  0.8622, Adjusted R-squared:  0.8586
## F-statistic: 238.3 on 10 and 381 DF,  p-value: < 2.2e-16
```

```
mpg_poly2 = lm(mpg~.-name-cylinders + log(weight) + log(acceleration) +
sqrt(displacement),data=Auto)
summary(mpg_poly2)
```

```
##
## Call:
## lm(formula = mpg ~ . - name - cylinders + log(weight) + log(acceleration) +
##     sqrt(displacement), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2104  -1.6665  -0.1085   1.5977  12.5231
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        290.113140  49.599189   5.849 1.06e-08 ***
## displacement         0.032477   0.028592   1.136 0.256720
## horsepower          -0.043782   0.013065  -3.351 0.000885 ***
## weight               0.006923   0.002226   3.110 0.002013 **
## acceleration         2.001283   0.468834   4.269 2.48e-05 ***
## year                 0.801707   0.044950  17.836  < 2e-16 ***
## origin               0.502973   0.262462   1.916 0.056064 .
## log(weight)        -34.848861   6.862200  -5.078 5.96e-07 ***
## log(acceleration)  -33.152402   7.671145  -4.322 1.98e-05 ***
## sqrt(displacement)  -1.043089   0.820337  -1.272 0.204311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.914 on 382 degrees of freedom
## Multiple R-squared:  0.8639, Adjusted R-squared:  0.8607
## F-statistic: 269.3 on 9 and 382 DF,  p-value: < 2.2e-16
```

- Both `mpg_poly` and `mpg_poly2` models see an increase in R^2 metric from 0.82 to 0.86. There are differences in the statistical significance of some predictors between the transformed models and the multiple regression model in **(c)**.

10. (a)

```
#fix(Carseats)
carseats_lm = lm(Sales~Price+Urban+US,data=Carseats)
summary(carseats_lm)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## 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
```

(b)

- The intercept represents the number of car seats sold on average when all other predictors are disregarded.
- The `Price` coefficient is negative and so sales will fall by roughly 54 seats(0.054x1000)for every unit($1) increase in price.
- The `Urban=Yes` coeff is not statistically significant. The `US=Yes` coeff is 1.2, and this means an average increase in car seat sales of 1200 units when `US=Yes`(this predictor likely refers to the shop being in the USA).

(c)

Dummy variables used by R:

```
attach(Carseats)
contrasts(US)
```

```
##     Yes
## No    0
## Yes   1
```

```r
contrasts(Urban)
```

```
##     Yes
## No    0
## Yes   1
```

Equation:

$$Sales = 13.04\ + -0.05Price\ \ + -0.02Urban(Yes:1,No:0)\ \ + 1.20US(Yes:1,No:0)$$

(d)

Using all variables:

```r
carseats_all_lm = lm(Sales~.,data=Carseats)
summary(carseats_all_lm)
```

```
##
## Call:
## lm(formula = Sales ~ ., data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8692 -0.6908  0.0211  0.6636  3.4115
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.6606231  0.6034487   9.380  < 2e-16 ***
## CompPrice        0.0928153  0.0041477  22.378  < 2e-16 ***
## Income           0.0158028  0.0018451   8.565 2.58e-16 ***
## Advertising      0.1230951  0.0111237  11.066  < 2e-16 ***
## Population       0.0002079  0.0003705   0.561    0.575
## Price           -0.0953579  0.0026711 -35.700  < 2e-16 ***
## ShelveLocGood    4.8501827  0.1531100  31.678  < 2e-16 ***
## ShelveLocMedium  1.9567148  0.1261056  15.516  < 2e-16 ***
## Age             -0.0460452  0.0031817 -14.472  < 2e-16 ***
## Education       -0.0211018  0.0197205  -1.070    0.285
## UrbanYes         0.1228864  0.1129761   1.088    0.277
## USYes           -0.1840928  0.1498423  -1.229    0.220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.019 on 388 degrees of freedom
## Multiple R-squared:  0.8734, Adjusted R-squared:  0.8698
## F-statistic: 243.4 on 11 and 388 DF,  p-value: < 2.2e-16
```

- Null hypothesis can be rejected for `CompPrice`, `Income`, `Advertising`, `Price`, `ShelvelocGood`, `ShelvelocMedium` and `Age`.

(e)

```r
carseats_all_lm2 = lm(Sales~.-Education-Urban-US-Population,data=Carseats)
summary(carseats_all_lm2)
```

```
##
## Call:
## lm(formula = Sales ~ . - Education - Urban - US - Population,
##     data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7728 -0.6954  0.0282  0.6732  3.3292
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.475226   0.505005   10.84   <2e-16 ***
## CompPrice        0.092571   0.004123   22.45   <2e-16 ***
## Income           0.015785   0.001838    8.59   <2e-16 ***
## Advertising      0.115903   0.007724   15.01   <2e-16 ***
## Price           -0.095319   0.002670  -35.70   <2e-16 ***
## ShelveLocGood    4.835675   0.152499   31.71   <2e-16 ***
## ShelveLocMedium  1.951993   0.125375   15.57   <2e-16 ***
## Age             -0.046128   0.003177  -14.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.019 on 392 degrees of freedom
## Multiple R-squared:  0.872,  Adjusted R-squared:  0.8697
## F-statistic: 381.4 on 7 and 392 DF,  p-value: < 2.2e-16
```

(f)

- The RSE goes down from 2.47 **model (a)** to 1.02 **model (e)**. The R2 statistic goes up from 0.24 **(a)** to 0.872 **(e)** and the F-statistic goes up from 41.52 to 381.4.
- The statistical evidence clearly shows that **(e)** is a much better fit.

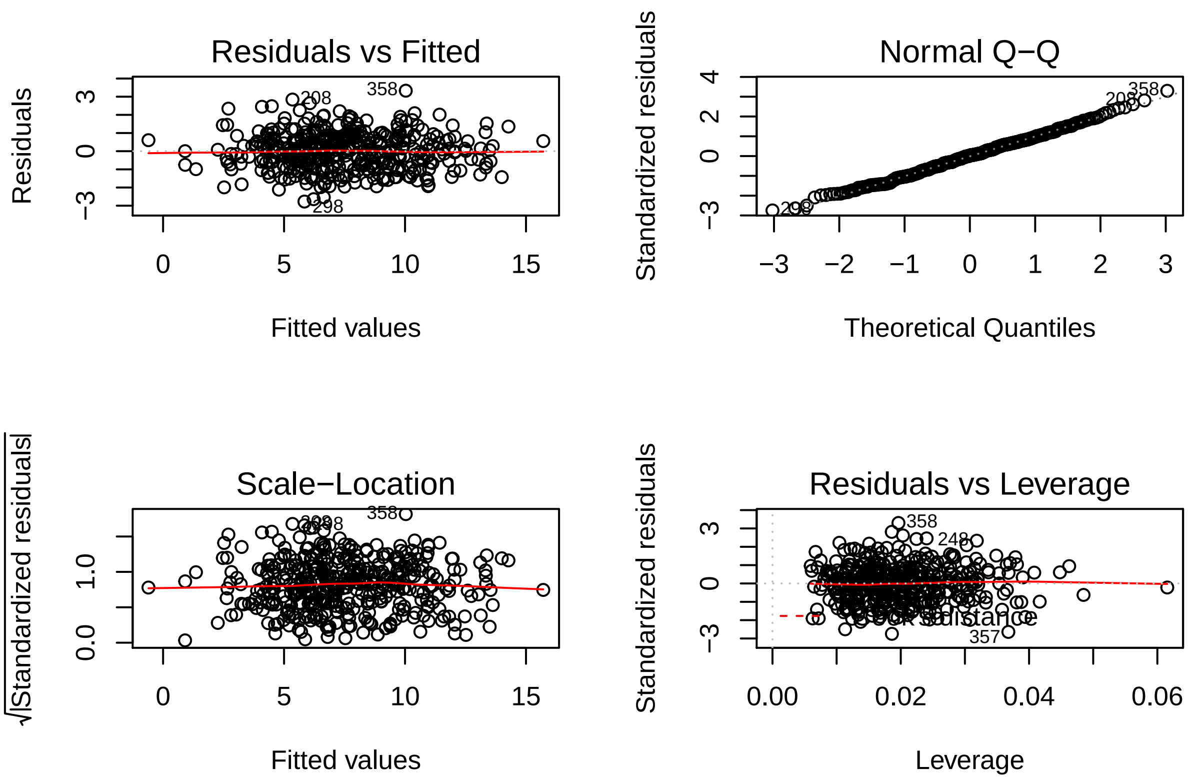
(g)

```r
confint(carseats_all_lm2)
```

```
##                       2.5 %      97.5 %
## (Intercept)      4.48236820  6.46808427
## CompPrice        0.08446498  0.10067795
## Income           0.01217210  0.01939784
## Advertising      0.10071856  0.13108825
## Price           -0.10056844 -0.09006946
## ShelveLocGood    4.53585700  5.13549250
## ShelveLocMedium  1.70550103  2.19848429
## Age             -0.05237301 -0.03988204
```

(h)

```r
par(mfrow=c(2,2))
plot(carseats_all_lm2)
```



- The residuals v fitted values chart doesn't show any distinct shape, so the model appears to be a good fit to the data.
- There appears to be some outliers. We can check as before by using studentized residuals. Observation 358 appears to an outlier.

```r
    rstudent(carseats_all_lm2)[which(rstudent(carseats_all_lm2)>3)]
```

```
##     358
## 3.34075
```

- There appears to be one high leverage observation.

```r
hatvalues(carseats_all_lm2)[order(hatvalues(carseats_all_lm2), decreasing = T)][1]
```

```
##        311
## 0.06154635
```

11. (a)

```r
set.seed (1)
x=rnorm (100)
y=2*x+rnorm (100)


lm.fit = lm(y~x+0)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

```r
#plot(y~x+0)
#abline(lm.fit, lwd =3, col ="red ")
```

- The coefficient is 1.99 and the Std. Error is 0.11. The p-value means that the null hypothesis can be rejected, hence the coefficient is statistically significant.

(b)

```r
lm.fit2 = lm(x~y+0)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

```
#plot(x~y+0)
#abline(lm.fit2, lwd =3, col ="blue")
```

- The coef estimate = 0.39, Std.Error = 0.02, t-statistic = 18.73, p-value < 2e-16. The t-statistic and so the p-values are exactly the same as with y onto x.

(c)

- We observe the same value of t-statistic, and hence p-values for y onto x and x onto y regression. The regression line is exactly the same with only the axes being changed, so y = 2x + e can be expressed as x = (y-e)/2.

(d)

Assuming summation limits as being from i=1 to n, I will exclude them to make equations more legible.

$$t^2 = \frac{\hat{\beta}^2}{SE(\hat{\beta})^2} = \frac{\hat{\beta}^2}{\sum(y_i - x_i\hat{\beta})^2/(n-1)\sum\ x_i^2} = \frac{(n-1)\sum(x_i^2)\hat{\beta}^2}{\sum(y_i - x_i\hat{\beta})^2}$$

Replace $\hat{\beta}^2$ in the numerator and expanding the denominator:

$$t^2 = \frac{(n-1)\sum(x_iy_i)^2}{\sum(x_i)^2\sum(y_i - x_i\hat{\beta})^2} = \frac{(n-1)\sum(x_iy_i)^2}{\sum\ x_i^2\sum\ y_i^2 - \sum\ x_i^2\sum 2\hat{\beta}\ x_iy_i + \sum\ x_i^2\sum(\hat{\beta}\ x_i)^2}$$

Replace $\hat{\beta}^2$ in the denominator and simplifying:

$$t^2 = \frac{(n-1)\sum(x_iy_i)^2}{\sum\ x_i^2\sum\ y_i^2 - 2\sum\ (x_iy_i)^2 + \sum\ (x_iy_i)^2} = \frac{(n-1)\sum(x_iy_i)^2}{\sum\ x_i^2\sum\ y_i^2 - \sum\ (x_iy_i)^2}$$

Finally:

$$t = \frac{\sqrt{(n-1)}\sum(x_iy_i)}{\sqrt{\sum\ x_i^2\sum\ y_i^2 - \sum\ (x_iy_i)^2}}$$

Confirming the above equation in R:

```
eqn_top = sqrt(length(x)-1)*sum(x*y)
eqn_bottom = sqrt(sum(x^2)*sum(y^2) - sum(x*y)^2)
t_statistic = eqn_top/eqn_bottom
print(t_statistic)
```

```
## [1] 18.72593
```

(e)

- It is obvious from the final result in (d) that replacing x_i with y_i or vice-versa gives the same t-statistic.

(f)

```r
lm.fit3 = lm(y~x)
summary(lm.fit3)$coefficients[2,3]
```

```
## [1] 18.5556
```

```r
lm.fit4 = lm(x~y)
summary(lm.fit4)$coefficients[2,3]
```

```
## [1] 18.5556
```

- As can be seen, the t-statistic is the same for both regressions.

12. (a)

- For regression of y onto x: $\hat{\beta} = \sum_{i=1}^{n}(x_i y_i)/\sum_{i'=1}^{n}(x_i')^2$, and for regression of x onto y: $\hat{\beta}' = \sum_{i=1}^{n}(x_i y_i)/\sum_{i'=1}^{n}(y_i')^2$.
- The coefficients are equal when the denominators are the same: $\sum_{i'=1}^{n}(x_i')^2 = \sum_{i'=1}^{n}(y_i')^2$

(b)

- see Q11(a) and Q11(b).

(c)

```r
set.seed (1)
x1=rnorm(100)
y1=sample(x1) # Changes order of x1 whilst making sure sum(x1)=sum(x2)


# Regression fits.
lm.fit4 = lm(x1~y1+0)
lm.fit5 = lm(y1~x1+0)

# Coefficients
summary(lm.fit4)$coefficients[1,1]
```

```
## [1] -0.07767695
```

```r
summary(lm.fit5)$coefficients[1,1]
```

```
## [1] -0.07767695
```

- The coefficients are the same as expected.

13. (a) (b) (c) (d) (e) (f)

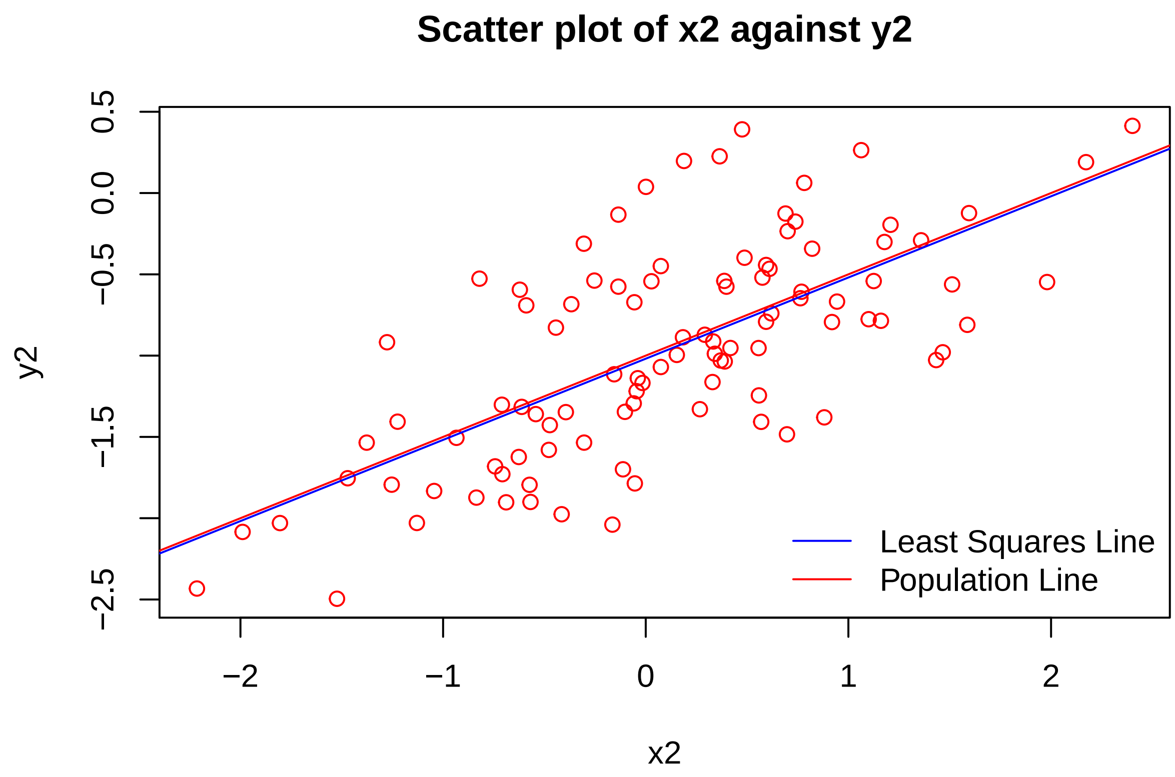
```r
set.seed(1)
x2 = rnorm(100, mean=0, sd=1)
eps = rnorm(100, mean=0, sd=0.5)
y2 = -1 +(0.5*x2) + eps
```

- Length of y2=100, $\beta_0 = -1$, $\beta_1 = 0.5$

```r
plot(y2~x2, main= 'Scatter plot of x2 against y2', col='red')

# Linear regression line.
lm.fit6 = lm(y2~x2)
summary(lm.fit6)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x2           0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

```r
abline(lm.fit6, lwd=1, col ="blue")

# Population regression line and legends.
abline(a=-1,b=0.5, lwd=1, col="red")
legend('bottomright', bty='n', legend=c('Least Squares Line', 'Population Line'),
       col=c('blue','red'), lty = c(1, 1))
```



- A positive linear relationship exists between x2 and y2, with added variance introduced by the error terms.
- $\hat{\beta}_0 = -1.018$ and $\hat{\beta}_1 = 0.499$.The regression estimates are very close to the true values: $\beta_0 = -1$, $\beta_1 = 0.5$. This is further confirmed by the fact that the regression and population lines are very close to each other. P-values are near zero and F-statistic is large so null hypothesis can be rejected.

(g)

```
# Polynomial regression
lm.fit7 = lm(y2~x2+I(x2^2))
summary(lm.fit7)
```

```
##
## Call:
## lm(formula = y2 ~ x2 + I(x2^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x2           0.50858    0.05399   9.420  2.4e-15 ***
## I(x2^2)     -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

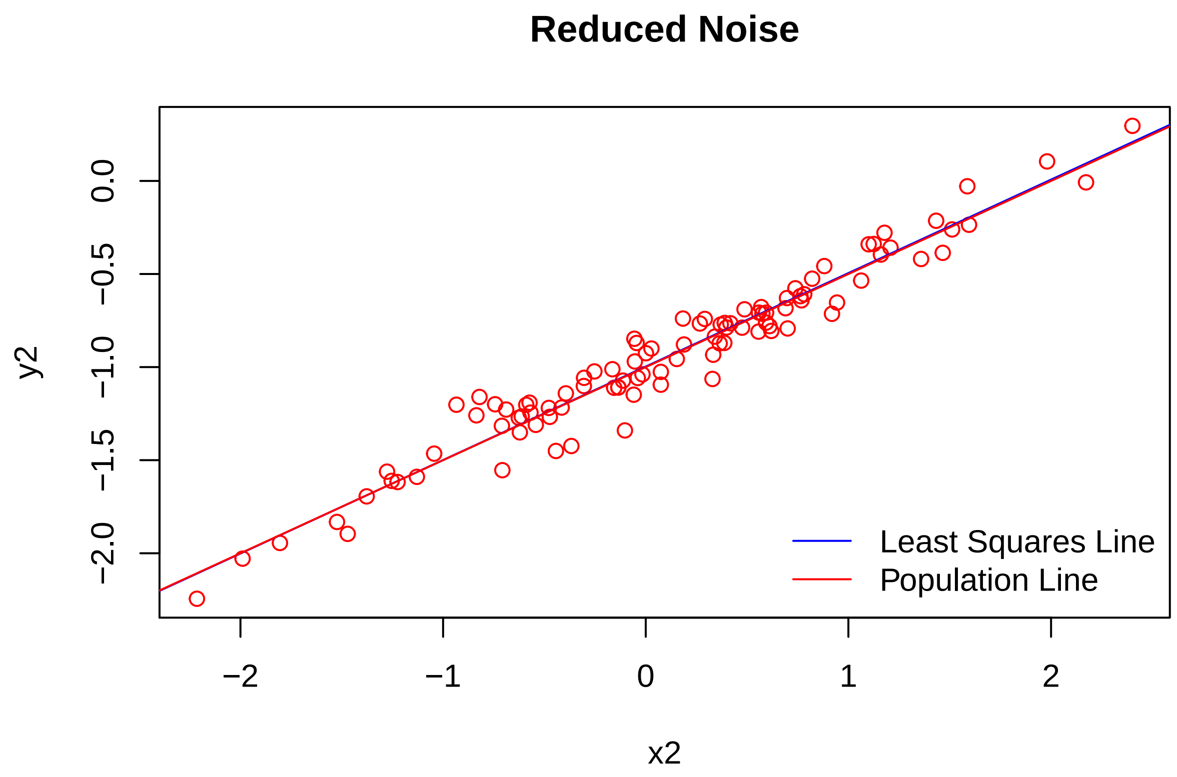
- The quadratic term does not improve the model fit. The F-statistic is reduced, and the p-value for the squared term is higher than 0.05 and shows that it isn't statistically significant.

(h)

```
eps = rnorm(100, mean=0, sd=sqrt(0.01))
y2 = -1 +(0.5*x2) + eps

plot(y2~x2, main='Reduced Noise', col='red')
lm.fit7 = lm(y2~x2)
summary(lm.fit7)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.291411 -0.048230 -0.004533  0.064924  0.264157
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99726    0.01047  -95.25   <2e-16 ***
## x2           0.50212    0.01163   43.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1039 on 98 degrees of freedom
## Multiple R-squared:  0.9501, Adjusted R-squared:  0.9495
## F-statistic:  1864 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
abline(lm.fit7, lwd=1, col ="blue")

abline(a=-1,b=0.5, lwd=1, col="red")
legend('bottomright', bty='n', legend=c('Least Squares Line', 'Population Line'),
       col=c('blue','red'), lty = c(1, 1))
```



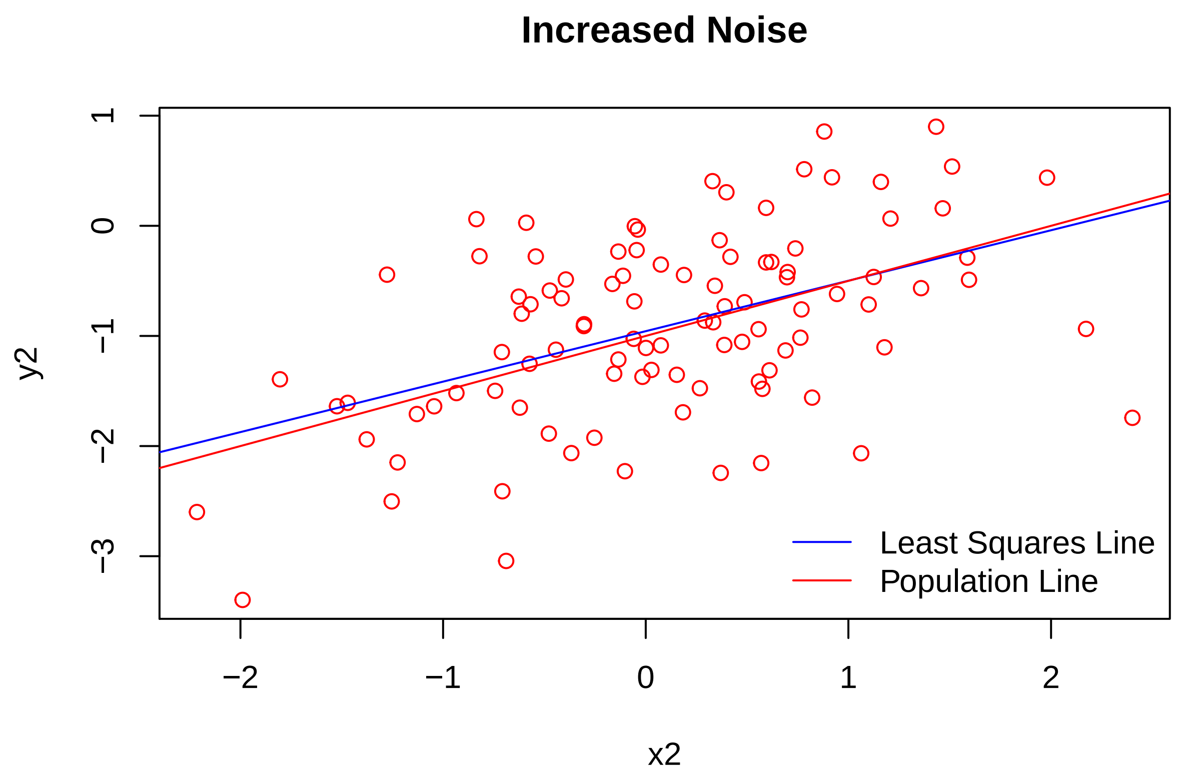
- The points are closer to each other, the RSE is lower, R2 and F-statistic are much higher than with variance of 0.25. The linear regression and population lines are very close to each other as noise is reduced, and the relationship is much more linear.

(i)

```
eps = rnorm(100, mean=0, sd=sqrt(0.5625))
y2 = -1 +(0.5*x2) + eps

plot(y2~x2, main='Increased Noise', col='red')
lm.fit8 = lm(y2~x2)
summary(lm.fit8)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.88719 -0.40893 -0.02832  0.50466  1.40916
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.95675    0.07521 -12.721  < 2e-16 ***
## x2           0.45824    0.08354   5.485 3.23e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7466 on 98 degrees of freedom
## Multiple R-squared:  0.2349, Adjusted R-squared:  0.2271
## F-statistic: 30.09 on 1 and 98 DF,  p-value: 3.227e-07
```

```
abline(lm.fit8, lwd=1, col ="blue")

abline(a=-1,b=0.5, lwd=1, col="red")
legend('bottomright', bty='n', legend=c('Least Squares Line', 'Population Line'),
       col=c('blue','red'), lty = c(1, 1))
```



- The points are more spread out and so the relationship is less linear. The RSE is higher, the R2 and F-statistic are lower than with variance of 0.25.

(j)

```
# 95% confidence intervals for original, less noise and more noise datasets respectively.
confint(lm.fit6)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1150804 -0.9226122
## x2           0.3925794  0.6063602
```

```
confint(lm.fit7)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0180413 -0.9764850
## x2           0.4790377  0.5251957
```

```
confint(lm.fit8)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1060050 -0.8074970
## x2           0.2924541  0.6240169
```

- Confidence interval values are narrowest for the lowest variance model, widest for the highest variance model and in-between these two for the original model.

14. (a) (b) (c)

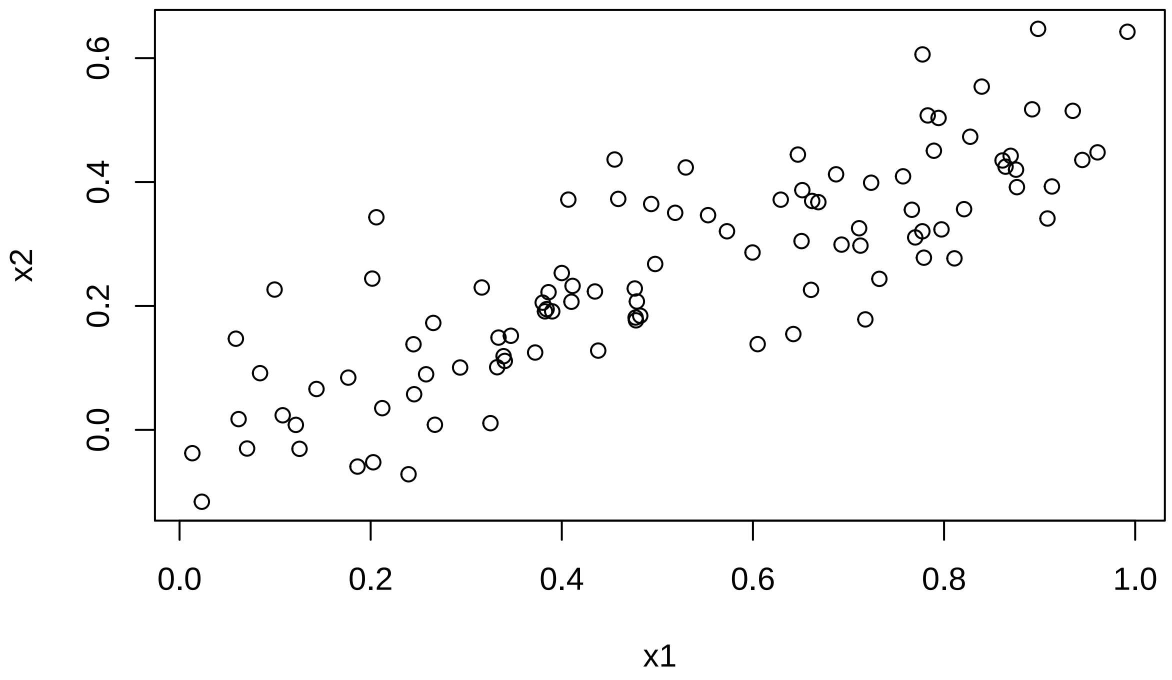
```
set.seed (1)
x1 = runif(100)
x2 = 0.5*x1+rnorm(100)/10
y = 2+2*x1+0.3*x2+rnorm(100)
```

- $Y = 2 + 2x_1 + 0.3x_2 + \epsilon$, where ϵ is a N(0,1) variable.
- $\beta_0 = 2$, $\beta_1 = 2$, $\beta_2 = 0.3$

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

```
## [1] 0.8351212
```

```
plot(x2~x1)
```



```r
lm.fit9 = lm(y~x1+x2)
summary(lm.fit9)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

- Coefficients from regression: $\hat{\beta}_0 = 2.13$, $\hat{\beta}_1 = 1.44$, $\hat{\beta}_2 = 1.00$. The Std. Error is high and increases from x1 to x2, and their respective regression coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ are inaccurate when compared to

$\beta_0 = 2$ and $\beta_1 = 2$ .The p-value for $\hat{\beta}_1$ is below 0.05 and so we can reject $H_0 : \hat{\beta}_1 = 0$. The p-value for $\hat{\beta}_2$ is much higher than 0.05 and so we cannot reject $H_0 : \hat{\beta}_2 = 0$.

(d)

```
lm.fit10 = lm(y~x1)
summary(lm.fit10)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

- RSE and R2 are similar to when using x1+x2, F-statistic is slightly greater.
- P-value is near zero for x1 and so $H_0 : \hat{\beta}_1 = 0$ can be rejected.

(e)

```
lm.fit11 = lm(y~x2)
summary(lm.fit11)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

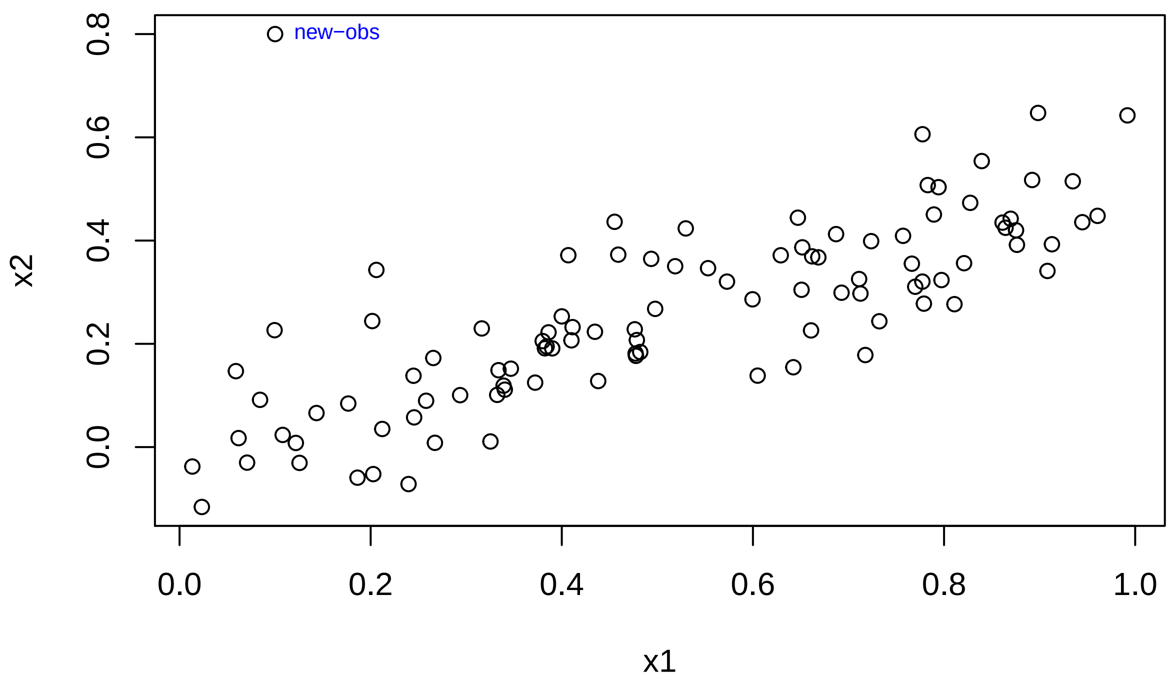
- P-value is near zero for x2 and so $H_0 : \hat{\beta}_1 = 0$ can be rejected.

(f)

- No, they do not contradict each other as the difference between **(c)** and **(e)** can be explained by the problem of collinearity. When using two variables that are highly collinear, the effect on the response of one variable can be masked by another. Collinearity also causes the standard error to increase - as can be seen the std. error of x1+x2 is greater than x1 or x2.

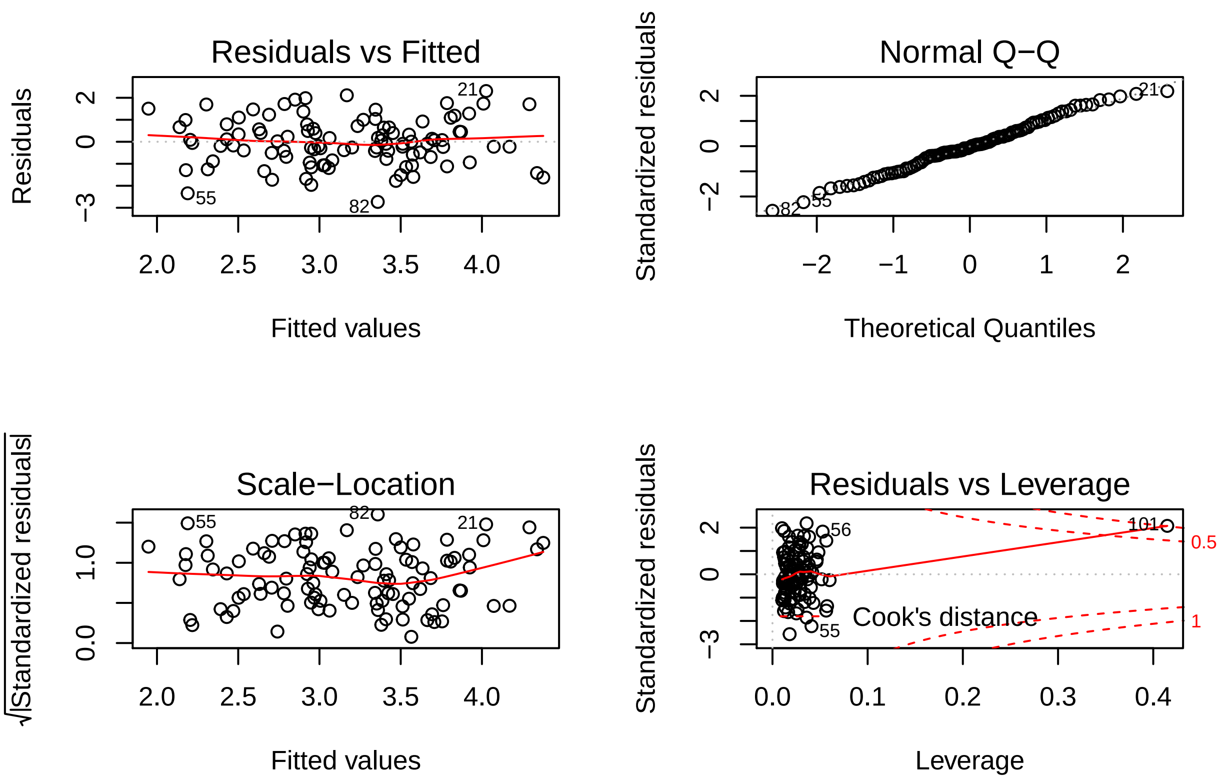
(g)

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

lm.fit9_g = lm(y~x1+x2)
lm.fit10_g = lm(y~x1)
lm.fit11_g = lm(y~x2)

plot(x1,x2)
text(x=0.1, y=0.8, labels="new-obs", pos=4, cex=.7, col="blue")
```



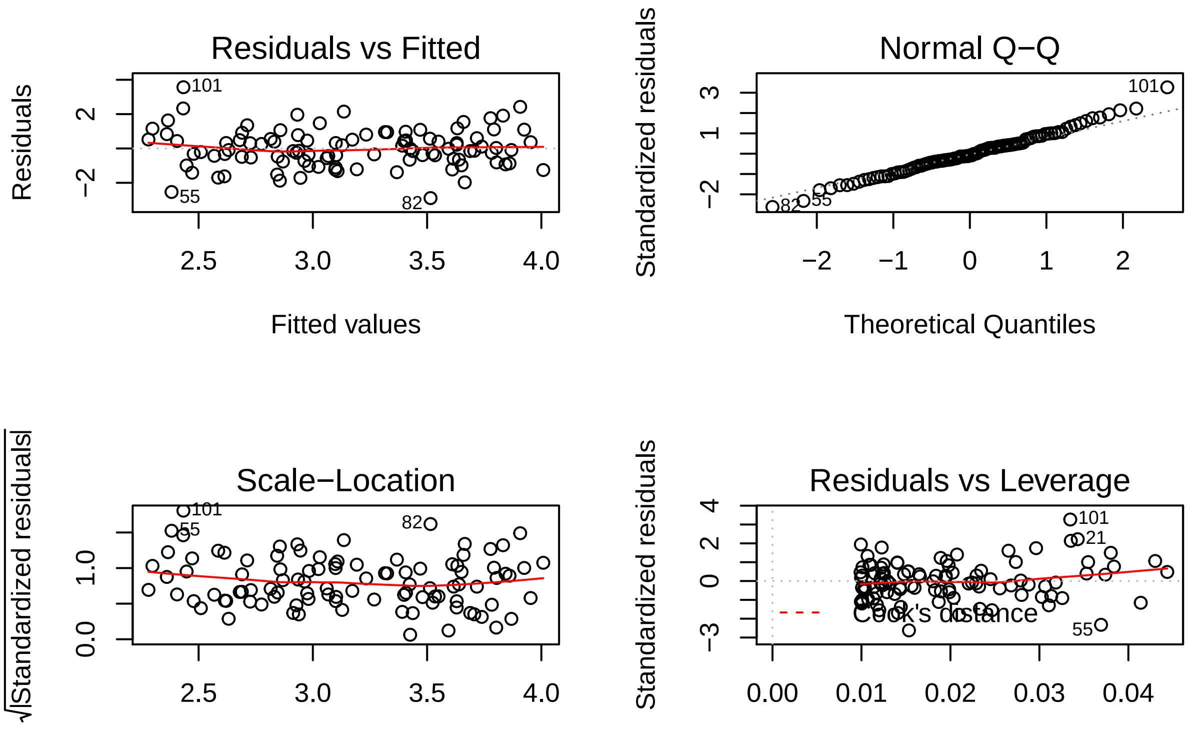
- The new point (101) appears to be an outlier. We can further investigate by using diagnostic plots for each model.

```r
par(mfrow=c(2,2))
plot(lm.fit9_g)
```



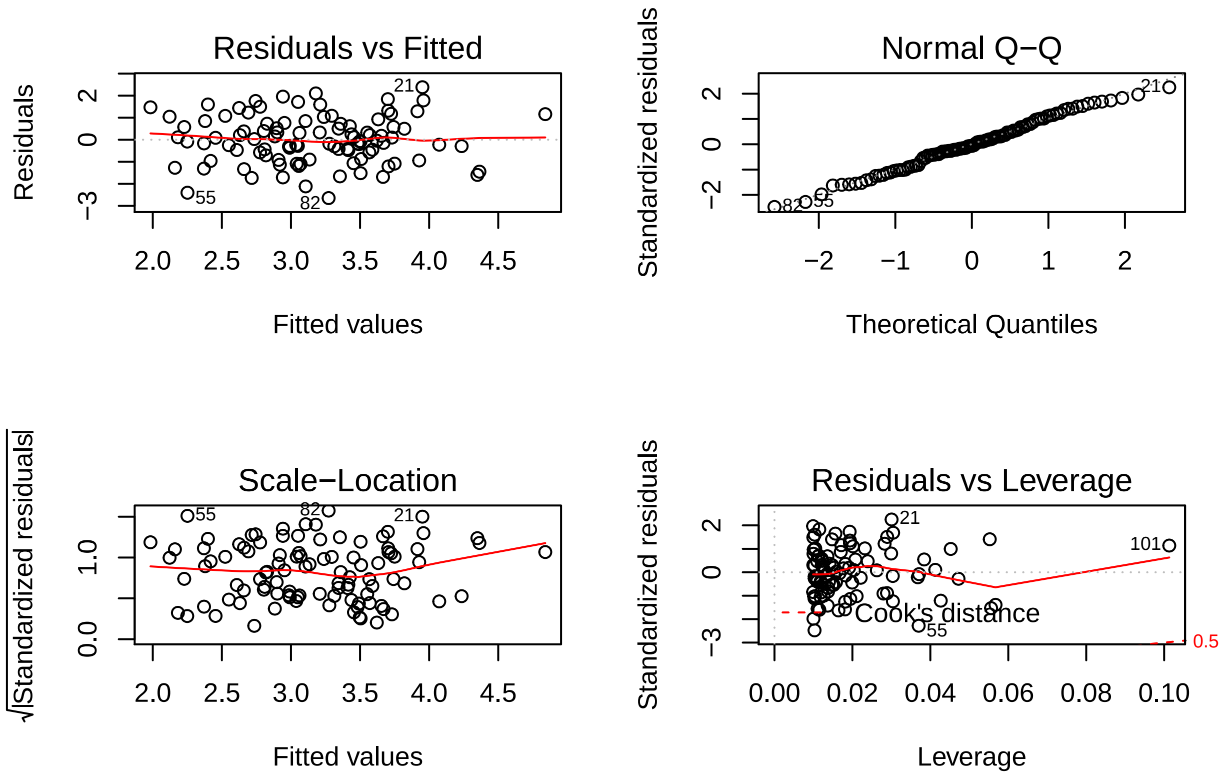
- Point 101 is high leverage but not an outlier.

```r
par(mfrow=c(2,2))
plot(lm.fit10_g)
```



- Point 101 is an outlier.

```r
par(mfrow=c(2,2))
plot(lm.fit11_g)
```



- Point 101 is high leverage.

15. (a)

```
library(MASS)
summary(Boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

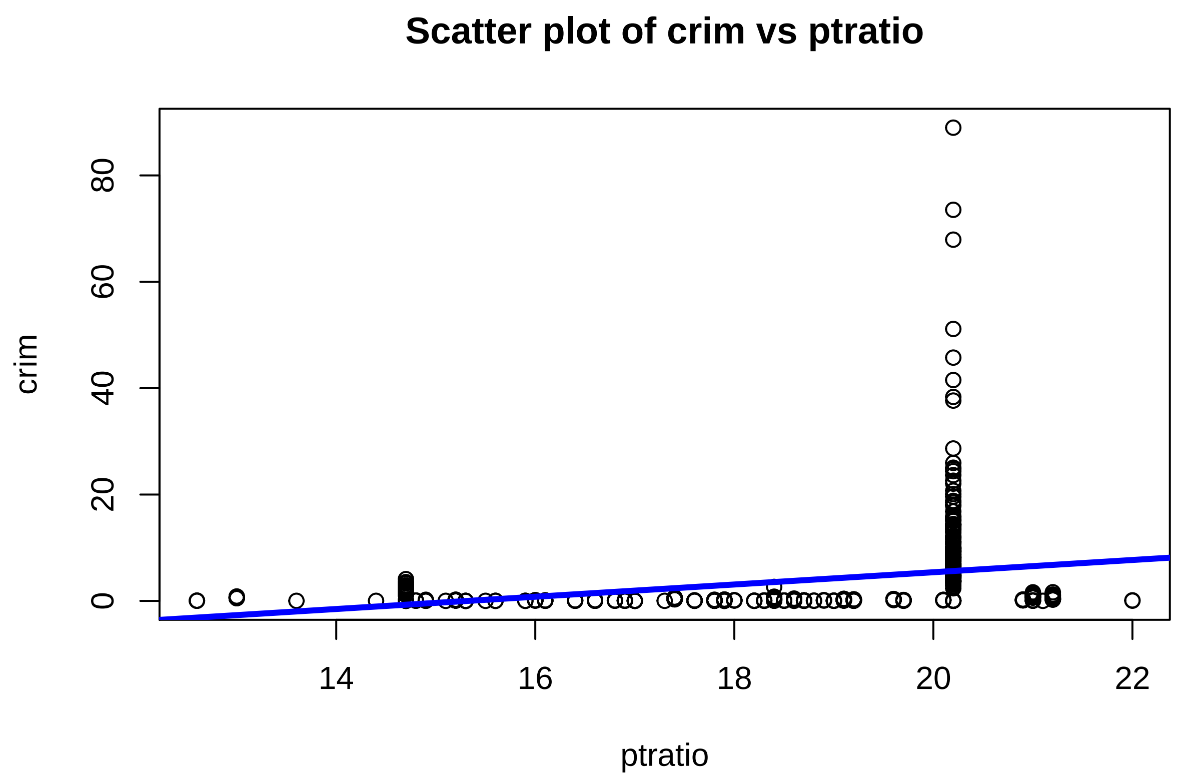
```r
#Linear regression of per capita crime onto each variable.
lm.zn = lm(crim~zn, data=Boston)
lm.indus = lm(crim~indus, data=Boston)
lm.chas = lm(crim~chas, data=Boston)
lm.nox = lm(crim~nox, data=Boston)
lm.rm = lm(crim~rm, data=Boston)
lm.age = lm(crim~age, data=Boston)
lm.dis = lm(crim~dis, data=Boston)
lm.rad = lm(crim~rad, data=Boston)
lm.tax = lm(crim~tax, data=Boston)
lm.ptratio = lm(crim~ptratio, data=Boston)
lm.black = lm(crim~black, data=Boston)
lm.lstat = lm(crim~lstat, data=Boston)
lm.medv = lm(crim~medv, data=Boston)
```

```r
# Dataframe with empty columns for p-values and coefficients.
df_pvalues = data.frame("p_values"= NA[1:13])
df_pvalues$coefficients = NA

# Change row name to predictor names.
row.names(df_pvalues) = names(Boston[2:14])

# For loop to extract and add to dataframe each p-value and coefficient.
for (i in 1: 13){
   lm_str = str2lang(paste("lm",row.names(df_pvalues)[i], sep="."))
   df_pvalues[i,1]= summary(eval(lm_str))$coefficients[2,4]
   df_pvalues[i,2]= summary(eval(lm_str))$coefficients[2,1]
   }

df_pvalues
```

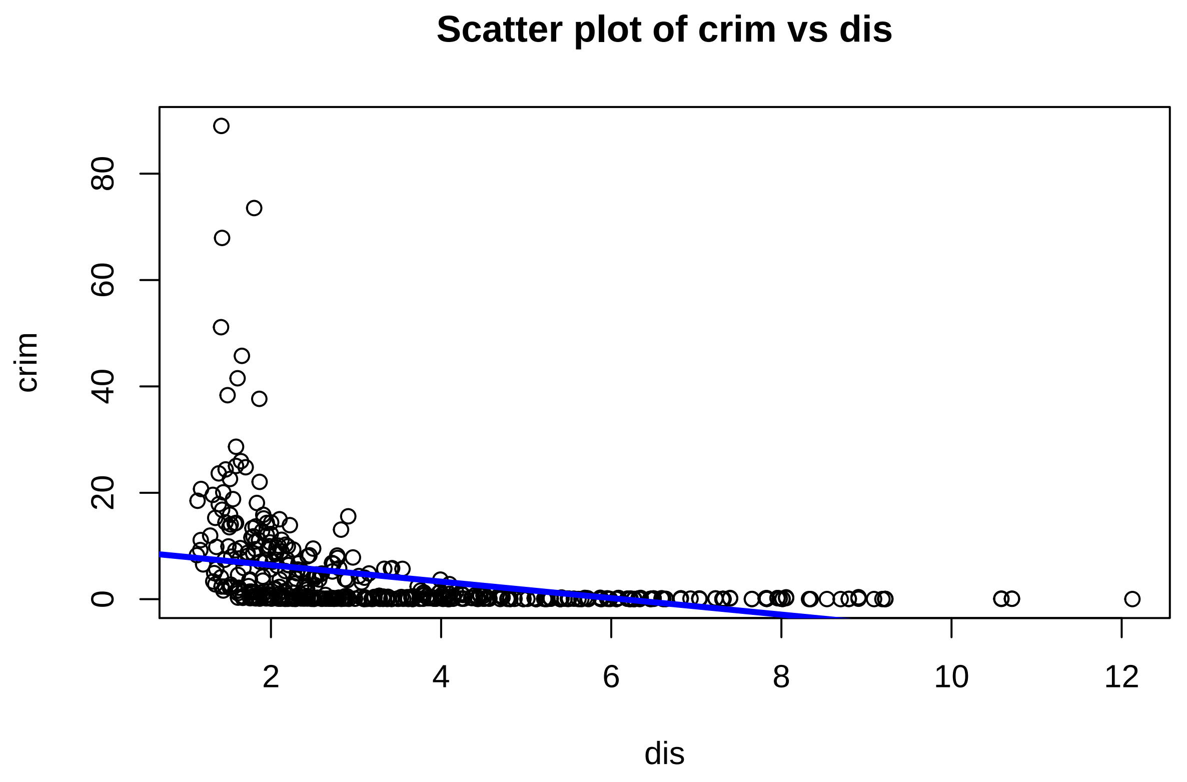
```
##             p_values coefficients
## zn      5.506472e-06  -0.07393498
## indus   1.450349e-21   0.50977633
## chas    2.094345e-01  -1.89277655
## nox     3.751739e-23  31.24853120
## rm      6.346703e-07  -2.68405122
## age     2.854869e-16   0.10778623
## dis     8.519949e-19  -1.55090168
## rad     2.693844e-56   0.61791093
## tax     2.357127e-47   0.02974225
## ptratio 2.942922e-11   1.15198279
## black   2.487274e-19  -0.03627964
## lstat   2.654277e-27   0.54880478
## medv    1.173987e-19  -0.36315992
```

- The p-values show that all predictors except `chas` are statistically significant.
- The coefficients show a positive linear relationship between `crim` and `ptratio` and a negative relationship between `crim` and `dis`. We should expect `crim` to increase as `ptratio` increases and decrease as `dis` increases. See charts below showing scatter plots with regression lines in each case.

```r
attach(Boston)
plot(ptratio,crim, main="Scatter plot of crim vs ptratio")
abline(lm.ptratio, lwd =3, col ="blue")
```



```r
plot(dis,crim, main="Scatter plot of crim vs dis")
abline(lm.dis, lwd =3, col ="blue")
```



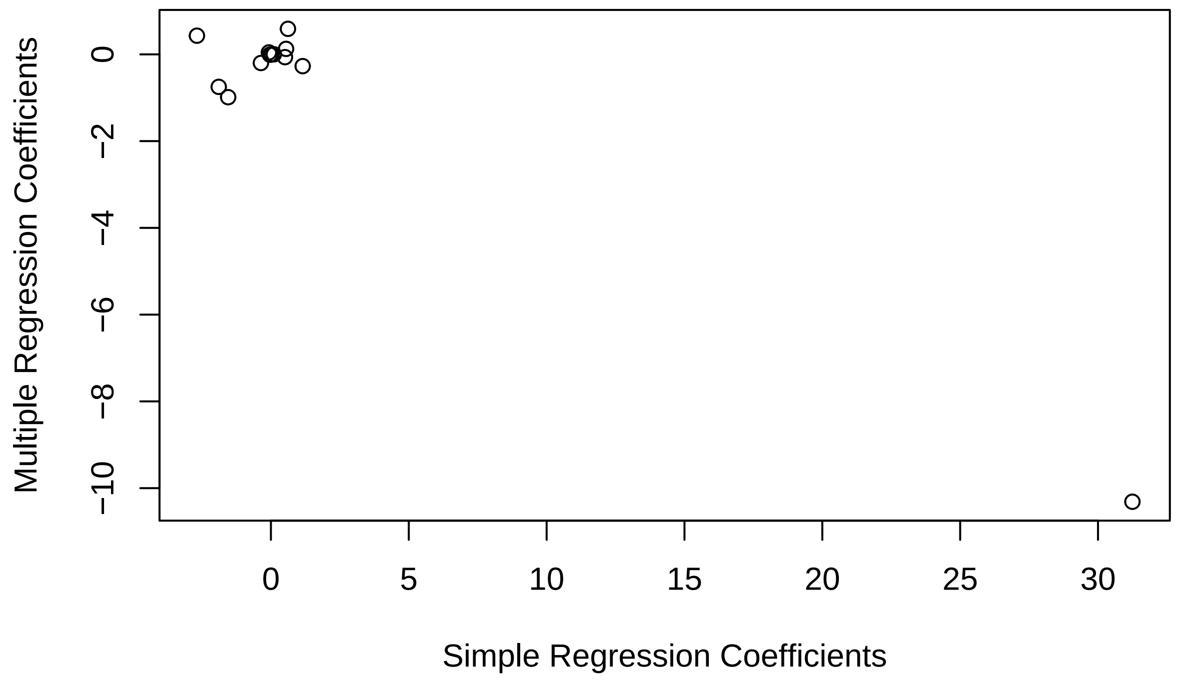
(b) (c)

```
# Regression using all predictors.
lm.fit_all = lm(crim~., data=Boston)
summary(lm.fit_all)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## 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 coefficient estimates and statistical significance for predictors have changed. For example, `ptratio` has a negative coefficient here, whereas it had a positive one in simple regression. In simple regression the coefficient of a predictor is calculated whilst ignoring all other predictors, and in multiple regression it is calculated whilst holding other predictors fixed. The difference in values between results from (a) and (b) isn't contradictory if the variables have collinearity - this was explored in Q14.
- Null hypothesis can be rejected for `zn`, `dis`, `rad`, `black` and `medv`.

```r
# Dataframe with coefficients from multiple regression and an empty column
# for coefficients from simple regression.
df_coefs = data.frame("multi_coefs"=summary(lm.fit_all)$coef[-1,1])
df_coefs$simple_coefs = NA

# Loop through each predictor, run linear regression and add values to dataframe.
for(i in row.names(df_coefs)){
    reg_model =  lm(crim~eval(str2lang(i)), data=Boston)
    df_coefs[row.names(df_coefs)==i, "simple_coefs"] = coef(reg_model)[2]
}


# Plot chart.
plot(df_coefs$simple_coefs, df_coefs$multi_coefs, xlab="Simple Regression Coefficients",
     ylab="Multiple Regression Coefficients")
```



(d)

```r
# Dataframe with columns for p-values of all polynomial terms.
df_poly = data.frame("variables"=names(Boston[2:14]))
df_poly$P_values_deg1 = NA
df_poly$P_values_deg2 = NA
df_poly$P_values_deg3 = NA

# Removed CHAS as it is a qualitative variable.
# It also caused error : 'degree' must be less than number of unique points.
df_poly = df_poly[-3,]
row.names(df_poly) <- NULL # Reset row numbers


# Loop through variables, run polynomial regression and add p-values to dataframe.
# Regression formula created from strings of each variable combined using paste.
for(i in 1:12){
    frmla1 = paste("poly(",paste(df_poly$variables[i],3,sep=","),")",sep="")
    frmla2 = paste("crim",frmla1,sep="~")
    frmla3 = as.formula(frmla2)
    poly_model = lm(frmla3, data=Boston)
    df_poly[i,2] = summary(poly_model)$coefficients[2,4]
    df_poly[i,3] = summary(poly_model)$coefficients[3,4]
    df_poly[i,4] = summary(poly_model)$coefficients[4,4]
}
```

```
df_poly
```

```
##    variables P_values_deg1 P_values_deg2 P_values_deg3
## 1         zn  4.697806e-06  4.420507e-03  2.295386e-01
## 2      indus  8.854243e-24  1.086057e-03  1.196405e-12
## 3        nox  2.457491e-26  7.736755e-05  6.961110e-16
## 4         rm  5.128048e-07  1.508545e-03  5.085751e-01
## 5        age  4.878803e-17  2.291156e-06  6.679915e-03
## 6        dis  1.253249e-21  7.869767e-14  1.088832e-08
## 7        rad  1.053211e-56  9.120558e-03  4.823138e-01
## 8        tax  6.976314e-49  3.665348e-06  2.438507e-01
## 9    ptratio  1.565484e-11  2.405468e-03  6.300514e-03
## 10     black  2.730082e-19  4.566044e-01  5.436172e-01
## 11     lstat  1.678072e-27  3.780418e-02  1.298906e-01
## 12      medv  4.930818e-27  2.928577e-35  1.046510e-12
```

- The degree 1 coefficients are all statistically significant - as expected.
- The degree 2 (squared fit) coefficients for zn, indus, nox, rm, age, dis, rad, tax, ptratio, lstat and medv are statistically significant.
- The degree 3 (cubic fit) coefficients of medv, dis, age, nox, ptratio and indus are statistically significant.
- The p-values support a quadratic fit for all variables except black, and also support a cubic fit for medv, dis, age, nox, ptratio and indus. Therefore, there is evidence of a non-linear relationship (either quadratic or cubic) for all variables except black.