Problem 5 Exercise 3.7

(a) Fit multiple regression model to predict sales

```
dat <- read.csv("Carseats.csv")
dat$Urban <- as.factor(dat$Urban)
dat$US <- as.factor(dat$US)
model <- lm(Sales ~ Price + Urban + US, data = dat)
summary(model)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = dat)
##
## 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) Provide an interpretation of each coefficient

The coefficient for Price is -0.054. That means for every unit increase in Price, the Sales will go down by 0.054. The coefficient for Urban is -0.023. That means if the store is urban, then sales will be 0.023 less than if it is not. The coefficient for US is 1.201. That means the sales is 1.201 higher if the store is in the U.S.

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

$$Sales = 13.043 + -0.054 * Price + -0.022 * Urban + 1.201 * US$$

For qualitative variables, Urban=1 if the store is in urban area, Urban=0 if it is not. US=1 if the store is in the U.S., US=0 if it is not.

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

Since the p-value for both Price and US are very close to zero, we can reject the null hypothesis, and say that it not equal to zero at confidence level 0.05.

(e) Fit a smaller model

```
model2 <- lm(Sales ~ Price + US, data = dat)
summary(model2)
```

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

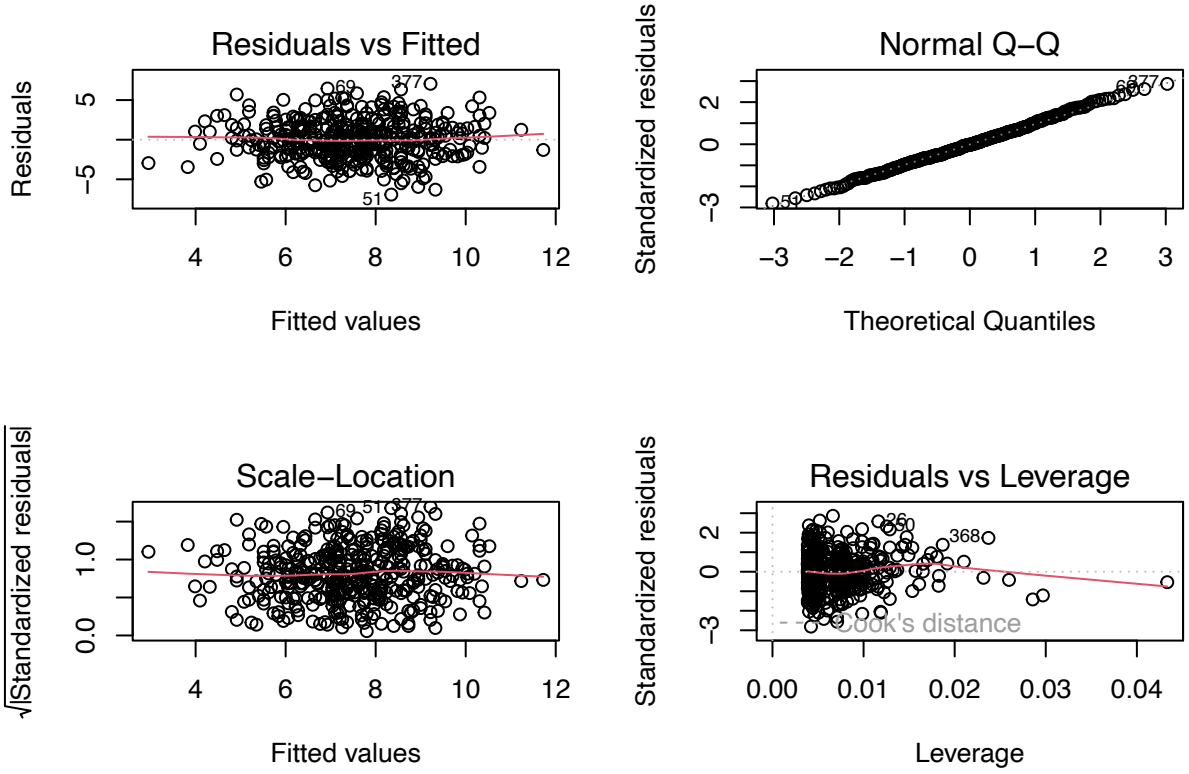
The adjust-R^2 for (a) is 0.2335, for (e) is 0.2354. So, the reduced model from part (e) is slightly better.

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

```
confint(model2)
```

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

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



In the residuals vs leverage plot, there are no outliers, all of them are less than 3. However, there are leverage points, because (p+1)/n=3/400=0.0075, we see that there are a few leverage points, but it is not very influential because the Cook's distance is lower than 1.

Problem 6 LDA vs Logistic Regression

```
wine <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data", sep = ",",
names(wine) <- c('class', 'Alcohol', 'Malic', 'Ash',
                      'Alcalinity', 'Magnesium', 'Phenols',
                      'Flavanoids', 'Nonflavanoids',
                      'Proanthocyanins', 'Color', 'Hue',
                      'Dilution', 'Proline')
wine$class <- as.factor(wine$class)
```

1. Implement a Linear Discriminant Analysis (LDA) classifier using all thirteen features to predict the three classes. Evaluate the accuracy of the LDA classifier on the training sample. (4 points)

```
library(MASS)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
modelLDA <- lda(class ~ ., wine)
pred <- predict(modelLDA)$class
result <- confusionMatrix(pred, wine$class)
result$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   1.000000e+00   1.000000e+00   9.794892e-01   1.000000e+00   3.988764e-01
## AccuracyPValue  McnemarPValue
##   8.896633e-72            NaN
```

```
wine.lda.predict <- train(class ~ ., method = "lda", data = wine)
confusionMatrix(wine$class, predict(wine.lda.predict, wine))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 59  0  0
##          2  0 71  0
##          3  0  0 48
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9795, 1)
##     No Information Rate : 0.3989
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000
## Prevalence             0.3315   0.3989   0.2697
## Detection Rate         0.3315   0.3989   0.2697
## Detection Prevalence   0.3315   0.3989   0.2697
## Balanced Accuracy      1.0000   1.0000   1.0000
```

The accuracy is 100%.

2. Repeat the same procedure for a multiclass logistic regression model

```
library(nnet)
modelmulti <- multinom(class ~., data = wine)
```

```
## # weights:  45 (28 variable)
## initial  value 195.552987
## iter  10 value 25.707822
## iter  20 value 4.439185
## iter  30 value 0.068485
## iter  40 value 0.000861
## final  value 0.000002
## converged
```

```
pred <- predict(modelmulti)
# result <- confusionMatrix(pred, wine$class)
# result$overall
mean(pred == wine$class)
```

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

3. The two methods both had 100% accuracy. Both methods are linear classifiers, we think that the classes are well separated linearly.