STATS 202: Data Mining and Analysis

Instructor: Linh Tran

HOMEWORK # 1

Due date: July 5, 2021

Stanford University

Introduction

Homework problems are selected from the course textbook: *An Introduction to Statistical Learning*.

Problem 1 (4 points)

Chapter 2, Exercise 2 (p. 52).

(a) Responses in this scenario is CEO salary, which is a quantitative variable; hence this is a **Regression** problem. Since the problem requires us to understand which factors affect CEO salaray we are interested in **inference**.

- $n = 500$
- $p =$ {profit, number of employs, industry}

(b) Responses in this scenario are success or failure, which is a categorical variable; hence this is a **Classification** problem. Since the problem requires us to predict if the new product launch will be a success or failure, we are interested in **prediction**.

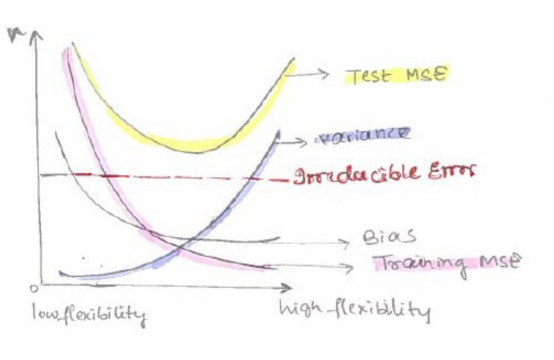
- $n = 20$
- $p =$ {price charged for the product, marketing budget, competition price, and 10 other variables not mentioned in the scenario description}

(c) Responses in this scenario are % changes in the US dollar, which is a quantitative variable; hence this is a **Regression** problem. Since the problem requires us to predict % change in the dollar in relation to the changes in the different world markets, we are interested in **prediction**.

- $n = 52$
- $p =$ {% change in the US market, % change in the British market, % change in the German market}

Problem 2 (4 points)

Chapter 2, Exercise 3 (p. 52).



- **Irreducible error**: Remains constant, as it is not affected by the flexibility of the statistical model.
- **Bias**: Starts off high and monotonically reduces as the flexibility of the statistical model increases because as the flexibility of the model increases, it fits the data better reducing the bias.
- **Variance**: Pretty low for restrictive statistical models and increases monotonically as the flexibility increases, as the model will follow the data closely resulting in overfitting.
- **Training MSE**: Typically high for restrictive models and starts decreasing as the model flexibility increases, as it follows the data closely and generates better (maybe even perfect) fits.
- **Test MSE**: Based on bias-variance decomposition, we know $MSE_{test}(x_0) = Var(\hat{f}_n(x_0)) + Bias(\hat{f}_n(x_0))^2 + Var(\epsilon_0)$. Based on this, test MSE is typically high for less flexible statistical methods, since though the variance is low, the bias component is much higher. It monotonically decreases as the flexibility increases to a certain level, and then gradually increases with models that have high flexibility. This is because as the flexibility of the statistical model increases, even though the bias is low, the variance component becomes very high.

Problem 3 (4 points)

Chapter 2, Exercise 7 (p. 53).

(a) In Cartesian coordinates, if $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ are two points in Euclidean n-space, then the distance $d(\mathbf{p}, \mathbf{q})$ from $\mathbf{p}$ to $\mathbf{q}$ (or $\mathbf{q}$ to $\mathbf{p}$) is given by:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (1)$$

With this equation, the Euclidean distance of $X_1 = X_2 = X_3 = 0$ with the training data is

Obs.	$d(\mathbf{p}, \mathbf{q})$	Y
1	$\sqrt{(0-0)^2 + (0-3)^2 + (0-0)^2} = 3$	Red
2	$\sqrt{(0-2)^2 + (0-0)^2 + (0-0)^2} = 2$	Red
3	$\sqrt{(0-0)^2 + (0-1)^2 + (0-3)^2} \approx 3.16$	Red
4	$\sqrt{(0-0)^2 + (0-1)^2 + (0-2)^2} \approx 2.24$	Green
5	$\sqrt{(0+1)^2 + (0-0)^2 + (0-1)^2} \approx 1.41$	Green
6	$\sqrt{(0-1)^2 + (0-1)^2 + (0-1)^2} \approx 1.73$	Red

Table 1: Euclidean distances between each observation and the test point.

(b) **Green**.

- **Justification**: The KNN estimator tries to find the K nearest elements and assigns the point to the class with the largest probability. With $K = 1$, the test point at $X_1 = X_2 = X_3 = 0$ is nearest to observation 5 with Euclidean distance (approximately) equal to 1.41. Since $K = 1$, we use only this observation, giving us $P(Green) = 1.0$.

(c) **Red**.

- **Justification**: With $K = 3$, the test point at $X_1 = X_2 = X_3 = 0$ is nearest to observations 2, 5, and 6, with Euclidean distances 2, 1.41, and 1.73, respectively. Using the labels from those points, the probability of Red is 2/3 and of Green is 1/3. Since Red has the highest probability, the test point will be predicted to belong to Red.

(d) The best value of K will be **small**, since the decision boundary is highly non-linear, i.e. the model is very flexible and therefore very closely follows the data. To produce a highly non-linear decision boundary using KNN, we need to choose a smaller value for K.

Problem 4 (4 points)

Chapter 10, Exercise 1 (p. 413).

(a) *Proof.* We start from the left side and add a 0 in disguise:

$$\begin{aligned}
\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 &= \frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj}+\bar{x}_{kj}-x_{i'j})^2 \\
&= \frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2+2(x_{ij}-\bar{x}_{kj})(\bar{x}_{kj}-x_{i'j})+(\bar{x}_{kj}-x_{i'j})^2 \\
&= \frac{2}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2+(x_{ij}-\bar{x}_{kj})(\bar{x}_{kj}-x_{i'j}) \\
&\quad \text{since } i \text{ and } i' \text{ are just indices} \\
&= \frac{2}{|C_k|}\sum_{i\in C_k}\sum_{i'\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2+(x_{ij}-\bar{x}_{kj})(\bar{x}_{kj}-x_{i'j}) \\
&= \frac{2|C_k|}{|C_k|}\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2+\frac{2}{|C_k|}\sum_{j=1}^{p}\sum_{i\in C_k}(x_{ij}-\bar{x}_{kj})\sum_{i'\in C_k}(\bar{x}_{kj}-x_{i'j}) \\
&\quad \text{since } \sum_{i\in Y} x = |Y|x \\
&= 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2 \\
&\quad \text{since } \sum_{i'\in C_k}(\bar{x}_{kj}-x_{i'j}) = 0
\end{aligned}$$

□

(b) From (a), we know that the sum of squared deviations between observations and the centroid is less than the average sum of squared deviations between the observatiosn within a cluster. In K-means, during each step we assign the observations to the nearest centroid, thus reducing the sum of squared deviations each time, which means each time we assign observations to the nearest centroid we are minimizing the dissimilarity (WSS) between observations within the cluster, i.e. minimizing the objective of the K-means algororithm.

Let $C_k^{(t)}$ and $\bar{x}_k^{(t)}$denote the iterates at time t. Step 2(b) of the algorithm (pg. 388) obeys the following bound:

$$\sum_{i,i'\in C_k^{(t)}}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k^{(t)}}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj}^{(t)})^2 \geq 2\sum_{i\in C_k^{(t+1)}}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj}^{(t)})^2$$

In Step 2(a) the cluster means for each feature are the constants that minimize the sum-of-squared deviations. Therefore,

$$2\sum_{i\in C_k^{(t+1)}}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj}^{(t)})^2 \geq 2\sum_{i\in C_k^{(t+1)}}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj}^{(t+1)})^2 = \sum_{i,i'\in C_k^{(t+1)}}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\ ,.$$

Thus,

$$\sum_{i,i'\in C_k^{(t)}}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 \geq \sum_{i,i'\in C_k^{(t+1)}}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\,.$$

Problem 5 (4 points)

Chapter 10, Exercise 2 (p. 413).

(a) **First iteration**: As seen in the Table a below, the shortest distance is between points 1 and 2. They will be clustered to $A(1,2)$ first at height 0.3.

Linkage	Distance
(1,2)	0.3
(1,3)	0.4
(1,4)	0.7
(2,3)	0.5
(2,4)	0.8
(3,4)	0.45

Table 2: Complete linkage: first iteration distances.

Second iteration: As seen in Table a below, the shortest distance is between points 3 and 4. They will be clustered to $B(3,4)$ at height 0.45.

Linkage	Distance
(A,3) = max[(1,3), (2,3)]	0.5
(A,4) = max[(1,4), (2,4)]	0.8
(3,4)	0.45

Table 3: Complete linkage: second iteration distances.

Third iteration: As seen in Table a, clusters A(1,2) and B(3,4) are linked at height 0.8.

Linkage	Distance
(A,B) = max[(1,2), (1,4), (2,3), (2,4)]	0.8

Table 4: Complete linkage: third iteration distances.

The full dendogram is shown in Figure a.

(b) **First iteration**: As seen in Table b below, the shortest distance is between points 1 and 2. They will be clustered to $A(1,2)$ first at height 0.3.

Second iteration: As seen in Table b below, the shortest distance is between cluster $A(1,2)$ and 3. They will be clustered to $B(1,2,3)$ at height 0.4.

Third iteration: As seen in Table b, cluster B(1,2,3) and 4 are linked at height 0.45.

The full dendogram is shown in Figure b.

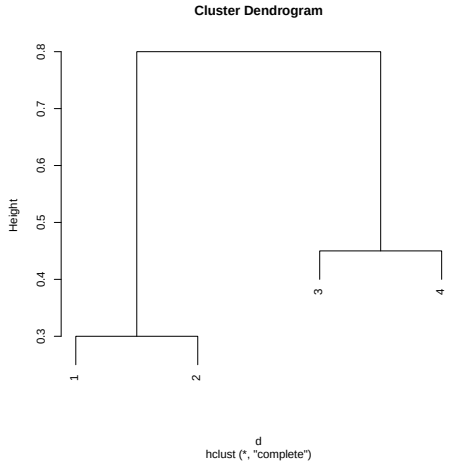


Figure 1: Complete linkage: resulting dendogram.

Linkage	Distance
(1,2)	0.3
(1,3)	0.4
(1,4)	0.7
(2,3)	0.5
(2,4)	0.8
(3,4)	0.45

Table 5: Single linkage: first iteration distances.

Linkage	Distance
(A,3) = min[(1,3), (2,3)]	0.4
(A,4) = min[(1,4), (2,4)]	0.7
(3,4)	0.45

Table 6: Single linkage: second iteration distances.

Linkage	Distance
(A,B) = min[(1,2), (1,4), (2,3), (2,4)]	0.45

Table 7: Single linkage: third iteration distances.

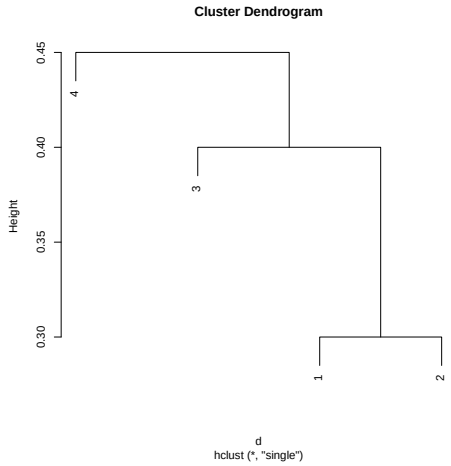


Figure 2: Single linkage: resulting dendogram.

(c) Cluster 1 = { 1, 2 }, Cluster 2 = { 3, 4 }

(d) Cluster 1 = { 1, 2, 3 }, Cluster 2 = { 4 }

(e) Figure e shows the plots with the leaves repositioned.

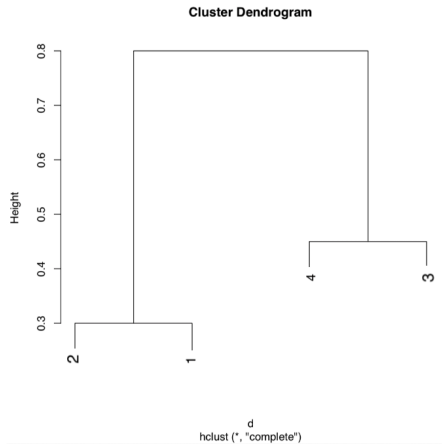


Figure 3: Complete linkage: resulting dendogram with leaves repositioned.

Problem 6 (4 points)

Chapter 10, Exercise 4 (p. 414).

(a) There is not enough information to tell. If we consider all the points within the 2 clusters are equi-distant from each other, then they fuse at the same height; if that is not the case then for complete linkage fusion will occur higher on the tree

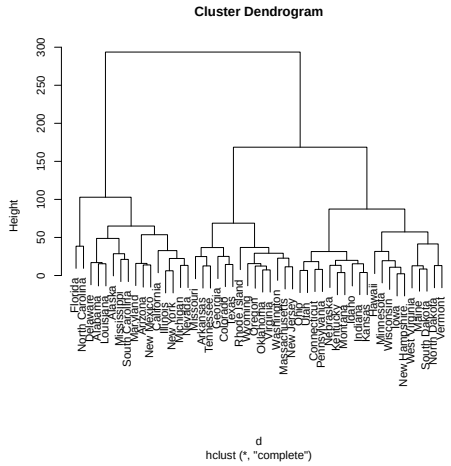
(b) Since the distance between the {5} and {6} doesn't change irrespective of the method used to cluster using hierarchical clustering, {5} and {6} fuse at the same height.

Problem 7 (4 points)

Chapter 10, Exercise 9 (p. 416).

```
data(USArrests)
```

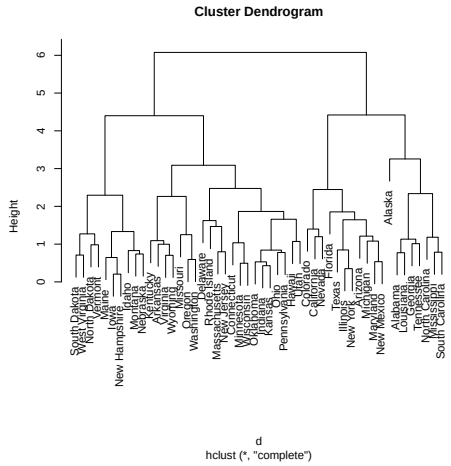
(a)
```
d <- dist(USArrests, method="euclidean")
h <- hclust(d, method="complete")
plot(h)
```



(b)
```
clusts <- cutree(h, 3)
for(i in 1:3) {
  cat(paste('--- States in cluster', i, '---\n'))
  print(names(clusts[clusts==i]))
  cat('\n')
}
```

```
## --- States in cluster 1 ---
##  [1] "Alabama"        "Alaska"         "Arizona"        "California"
##  [5] "Delaware"       "Florida"        "Illinois"       "Louisiana"
##  [9] "Maryland"       "Michigan"       "Mississippi"    "Nevada"
## [13] "New Mexico"     "New York"       "North Carolina" "South Carolina"
##
## --- States in cluster 2 ---
##  [1] "Arkansas"      "Colorado"      "Georgia"       "Massachusetts"
##  [5] "Missouri"      "New Jersey"    "Oklahoma"      "Oregon"
##  [9] "Rhode Island"  "Tennessee"     "Texas"         "Virginia"
## [13] "Washington"    "Wyoming"
##
## --- States in cluster 3 ---
##  [1] "Connecticut"   "Hawaii"        "Idaho"         "Indiana"
##  [5] "Iowa"          "Kansas"        "Kentucky"      "Maine"
##  [9] "Minnesota"     "Montana"       "Nebraska"      "New Hampshire"
## [13] "North Dakota"  "Ohio"          "Pennsylvania"  "South Dakota"
## [17] "Utah"          "Vermont"       "West Virginia" "Wisconsin"
```

(c)
```
d <- dist(scale(USArrests), method="euclidean")
h <- hclust(d, method="complete")
plot(h)
```




```
  clusts <- cutree(h, 3)
  for(i in 1:3) {
    cat(paste('--- States in cluster', i, '---\n'))
    print(names(clusts[clusts==i]))
    cat('\n')
  }
```

```
## --- States in cluster 1 ---
## [1] "Alabama"        "Alaska"         "Georgia"        "Louisiana"
## [5] "Mississippi"    "North Carolina" "South Carolina" "Tennessee"
##
## --- States in cluster 2 ---
##  [1] "Arizona"    "California" "Colorado"   "Florida"    "Illinois"
##  [6] "Maryland"   "Michigan"   "Nevada"     "New Mexico" "New York"
## [11] "Texas"
##
## --- States in cluster 3 ---
##  [1] "Arkansas"      "Connecticut"   "Delaware"      "Hawaii"
##  [5] "Idaho"         "Indiana"       "Iowa"          "Kansas"
##  [9] "Kentucky"      "Maine"         "Massachusetts" "Minnesota"
## [13] "Missouri"      "Montana"       "Nebraska"      "New Hampshire"
## [17] "New Jersey"    "North Dakota"  "Ohio"          "Oklahoma"
## [21] "Oregon"        "Pennsylvania"  "Rhode Island"  "South Dakota"
## [25] "Utah"          "Vermont"       "Virginia"      "Washington"
## [29] "West Virginia" "Wisconsin"     "Wyoming"
```

(d) Scaling the variables before preforming hierarchical clustering results in 3 clusters with different states in each one of them, compared to hierarchical clusters without scaling. In this case, it makes sense to scale the variables for 2 reasons:

(a) The unit of measurement of `UrbanPop` is different than that of `Rapge`, `Assault`, and `Murder`.

(b) The variance of `Assault` and `UrbanPop` is much higher, hence has a larger effect on computing dissimilarities between states.

```
  apply(USArrests, 2, var)
```

```
##     Murder    Assault   UrbanPop       Rape
##   18.97047 6945.16571  209.51878   87.72916
```

Problem 8 (4 points)

Chapter 3, Exercise 4 (p. 120).

(a) We expect training RSS for cubic regression to be lower than that of linear regression as the cubic regression over fits the data.

(b) Sicne the true relationship between X and Y is linear, linear regression provides a better fit than the fit provided by cubic regression. Hence the test set RSS will be lower for linear regression than for cubic regression.

(c) We expect training RSS for cubic regression to be lower than that of linear regression as the cubic regression provides better fit to the non-linearity in the data.

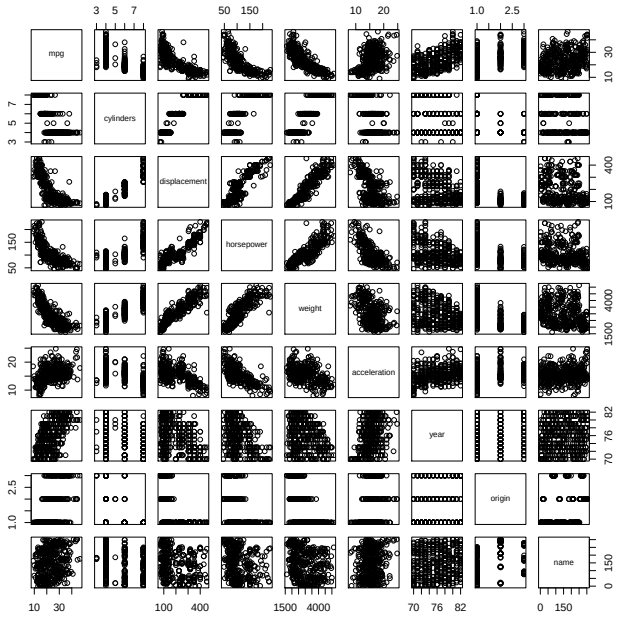
(d) Since the true relationship between X and Y is not linear and we don't know how far it is from being linear it is difficult to answer if the test set RSS for the linear model will be smaller than for cubic regression. If the true relationship is close to being linear, then the test set RSS for linear regression would be lower. If the true relationship is far from linear, then the test set RSS for cubic regression would be lower.

Problem 9 (4 points)

Chapter 3, Exercise 9 (p. 122). In parts (e) and (f), you need only try a few interactions and transformations.

```
library(ISLR)
data(Auto)
```

(a)
```
pairs(Auto)
```



It appears that `origin` is a categorical variable, but stored as a quantitative one. We will update to it be stored as a factor.

```
Auto$origin <- as.factor(Auto$origin)
```

(b)
```
cor(Auto[,-c(8:9)])
```

```
##                     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
##              acceleration       year
## mpg             0.4233285  0.5805410
## cylinders      -0.5046834 -0.3456474
## displacement   -0.5438005 -0.3698552
## horsepower     -0.6891955 -0.4163615
## weight         -0.4168392 -0.3091199
## acceleration    1.0000000  0.2903161
## year            0.2903161  1.0000000
```

(c)
```
lm_fit <- lm(mpg ~ . - name, data=Auto)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                 Estimate  Std. Error t value    Pr(>|t|)
## (Intercept)  -17.9546021   4.6769339  -3.839    0.000145 ***
## cylinders     -0.4897094   0.3212309  -1.524    0.128215
## displacement   0.0239786   0.0076533   3.133    0.001863 **
## horsepower    -0.0181835   0.0137086  -1.326    0.185488
## weight        -0.0067104   0.0006551 -10.243     < 2e-16 ***
## acceleration   0.0791030   0.0982185   0.805    0.421101
## year           0.7770269   0.0517841  15.005     < 2e-16 ***
## origin2        2.6300024   0.5664147   4.643 0.000004720 ***
## origin3        2.8532282   0.5527363   5.162 0.000000393 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242,Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

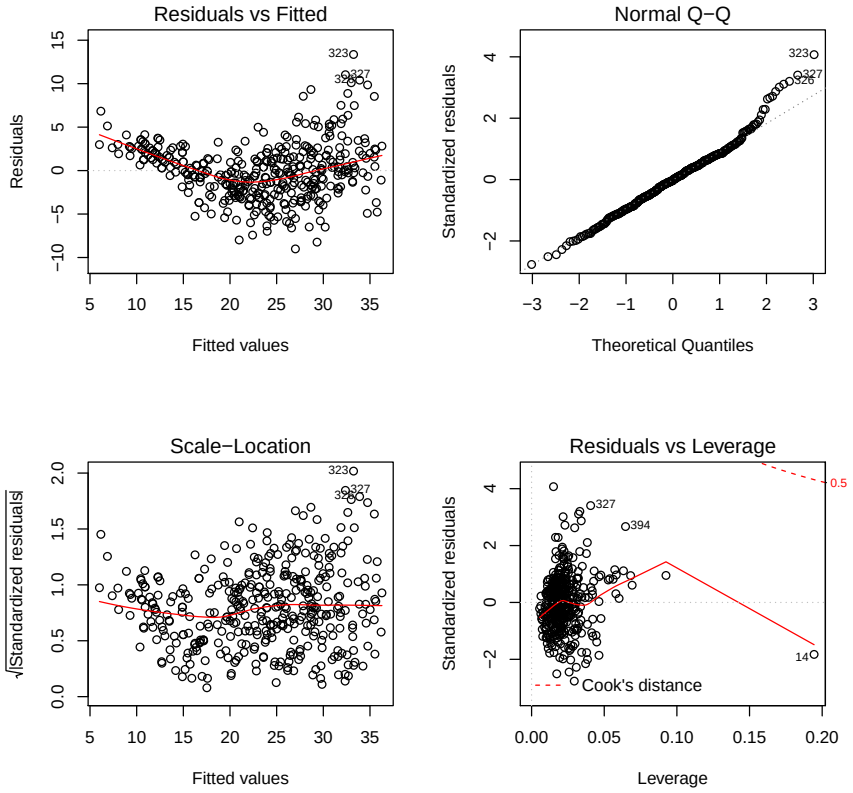
The summary shows us that

(a) The F-statistic is much greater than 1 and its associated p-value is smaller than 0.05. So at 0.05 significance level, we can say at least one of the predictors is related in estimating MPG.

(b) `displacement`, `weight`, `year`, `origin2`, and `origin3` have statistically significant relationships to the response `mpg`.

(c) On average `mpg` increases by 0.7770 units for each one unit increase in the year, holding all other predictors fixed.

(d)
```
par(mfrow=c(2,2))
plot(lm_fit)
```



There are a number of conclusions we can draw from analyzing the plots:

(a) The residual vs fitted plot exhibits a clear U-Shaped relationship, which provides a strong indication of non-linearity in the data.

(b) There is a clear funnel shap in the residual plot, indicating heteroscedasticity.

(c) The scale-location plot shows the absolute value of the studentized residuals (after taking the square-root in order to diminish potential skewness) plotted against the fitted values. Recall that absolute values larger than 3 are considered potential outliers. Three points (observations 323, 326, and 327) are above $\sqrt{3}$, indicating that they are possibly outliers.

(d) Based on the residuals vs leverage plot, observation 14 is a high leverage point. While (from the scale-location) plot, observaitons 323, 326, and 327 are outliers, they do not have high leverage.

(e) I've included pairwise interaction terms between every predictor being considered. From this, we see that `cylinders:acceleration`, `acceleration:year`, `acceleration:origin2`, `acceleration:origin3`, `year:origin2`, and `year:origin3` are statistically significant.

```
lm_fit_interations <- lm(mpg ~ .*., data=Auto[,-9])
summary(lm_fit_interations)
```

```
##
## Call:
## lm(formula = mpg ~ . * ., data = Auto[, -9])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6008 -1.2863  0.0813  1.2082 12.0382
##
## Coefficients:
##                               Estimate   Std. Error t value Pr(>|t|)
## (Intercept)                 44.01189478  51.46736258   0.855 0.393048
## cylinders                    3.30155433   8.18651422   0.403 0.686976
## displacement                -0.35289516   0.19737638  -1.788 0.074638 .
## horsepower                   0.53121951   0.33897387   1.567 0.117970
## weight                      -0.00325908   0.01819946  -0.179 0.857980
## acceleration                -6.04827289   2.14663070  -2.818 0.005109 **
## year                         0.48326326   0.59232770   0.816 0.415119
## origin2                    -35.16513846  12.60196356  -2.790 0.005547 **
## origin3                    -37.64639178  14.26129623  -2.640 0.008661 **
## cylinders:displacement      -0.00631556   0.00710566  -0.889 0.374707
## cylinders:horsepower         0.01451559   0.02457435   0.591 0.555109
## cylinders:weight             0.00057033   0.00090442   0.631 0.528709
## cylinders:acceleration       0.36581508   0.16713092   2.189 0.029261 *
## cylinders:year              -0.14465655   0.09652342  -1.499 0.134846
## cylinders:origin2           -0.72095509   1.08828549  -0.662 0.508100
## cylinders:origin3            1.22561429   1.00701105   1.217 0.224379
## displacement:horsepower     -0.00005407   0.00028609  -0.189 0.850212
## displacement:weight          0.00002659   0.00001455   1.828 0.068435 .
## displacement:acceleration   -0.00254656   0.00335560  -0.759 0.448415
## displacement:year            0.00454662   0.00244564   1.859 0.063842 .
## displacement:origin2        -0.03363689   0.04219705  -0.797 0.425902
## displacement:origin3         0.05375129   0.04144794   1.297 0.195527
## horsepower:weight           -0.00003407   0.00002955  -1.153 0.249743
## horsepower:acceleration     -0.00344504   0.00393684  -0.875 0.382122
## horsepower:year             -0.00642697   0.00389123  -1.652 0.099487 .
## horsepower:origin2          -0.00486943   0.05061262  -0.096 0.923408
## horsepower:origin3           0.02288515   0.06251629   0.366 0.714533
## weight:acceleration         -0.00006851   0.00023847  -0.287 0.774061
## weight:year                 -0.00008065   0.00021845  -0.369 0.712223
## weight:origin2               0.00227655   0.00268478   0.848 0.397037
## weight:origin3              -0.00449818   0.00348082  -1.292 0.197101
## acceleration:year            0.06141264   0.02546586   2.412 0.016390 *
## acceleration:origin2         0.92339618   0.26409862   3.496 0.000531 ***
## acceleration:origin3         0.71592928   0.32576288   2.198 0.028614 *
## year:origin2                 0.29322251   0.14437874   2.031 0.043005 *
## year:origin3                 0.31388902   0.14833731   2.116 0.035034 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.628 on 356 degrees of freedom
## Multiple R-squared:  0.8967,Adjusted R-squared:  0.8866
## F-statistic: 88.34 on 35 and 356 DF,  p-value: < 2.2e-16
```

(f) I've added 2 transformations: `displacement`^{2} and log(`year`). This increases the proportion of the variance explained from 82.42% to 86.81%, which is a huge improvement from the previous model with no transformations. Furthermore, the summary indicates that these transformations are statistically significant.

```
  lm_fit_transformation <- lm(mpg ~ . - name + I(displacement^2) + I(log(year)), data=Auto)
  summary(lm_fit_transformation)
```

```
##
## Call:
## lm(formula = mpg ~ . - name + I(displacement^2) + I(log(year)),
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6148 -1.5397  0.1025  1.4712 11.8630
##
## Coefficients:
##                       Estimate   Std. Error t value         Pr(>|t|)
## (Intercept)       2947.1344740  477.0733050   6.178 0.00000000167256 ***
## cylinders            0.8771402    0.3075073   2.852          0.00458 **
## displacement        -0.1062987    0.0149376  -7.116 0.00000000000554 ***
## horsepower          -0.0721834    0.0128461  -5.619 0.00000003711650 ***
## weight              -0.0034269    0.0006398  -5.356 0.00000014762213 ***
## acceleration        -0.0586677    0.0861950  -0.681          0.49651
## year                12.4545448    1.8859226   6.604 0.00000000013478 ***
## origin2              0.8646465    0.5348022   1.617          0.10676
## origin3              0.9103209    0.5260915   1.730          0.08438 .
## I(displacement^2)    0.0002040    0.0000218   9.359          < 2e-16 ***
## I(log(year))      -888.3045311  143.3074896  -6.199 0.00000000148141 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.872 on 381 degrees of freedom
## Multiple R-squared:  0.8681,Adjusted R-squared:  0.8646
## F-statistic: 250.7 on 10 and 381 DF,  p-value: < 2.2e-16
```

Problem 10 (4 points)

Chapter 3, Exercise 14 (p. 125).

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

(a) The form of the linear model is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \tag{2}$$

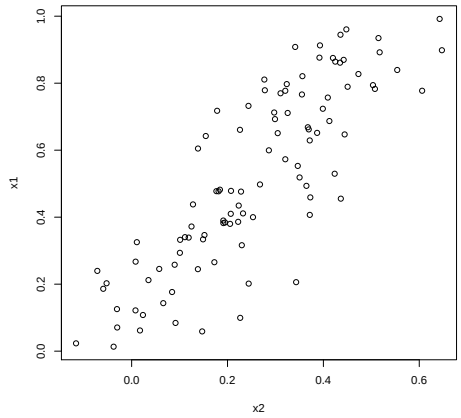
The regression coefficients are $\beta_0 = 2$, $\beta_1 = 2$, and $\beta_2 = 0.3$.

(b)
```
  est_cor <- cor(x1, x2)
  print(est_cor)
```

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

The correlation between x_1 and x_2 is 0.835.

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



(c)
```
fit_1 <- lm(y ~ x1 + x2)
summary(fit_1)
```

```
##
## 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: 0.00001164
```

x_1 and x_2 together explain 20.88% of the variability in y. The p-value associated with the F-statistic is less than 0.05, indicating that at least one of the predictors is related to the response y.

$\hat{\beta}_0 = 2.1305$, $\hat{\beta}_1 = 1.4396$, and $\hat{\beta}_2 = 1.0097$. While $\hat{\beta}_0$ and $\hat{\beta}_1$ are relatively close to β_0 and β_1, respectively, $\hat{\beta}_2$ is considerably greater than β_2.

We can reject $H_0 : \beta_1 = 0$, since the p-value associated with $\hat{\beta}_1$ is significant. However, we cannot reject $H_0 : \beta_2 = 0$, since the p-value associated with $\hat{\beta}_2$ is not significant.

(d)
```
fit_2 <- lm(y ~ x1)
summary(fit_2)
```

```
##
## 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: 0.000002661
```

The p-value associated with x_1 is significant and this model explains 20.24% of the variability of y. Both $\hat{\beta}_0$ and $\hat{\beta}_1$ are similar to β_0 and β_1, respectively.

We can reject $H_0 : \beta_1 = 0$, since the p-value associated with $\hat{\beta}_1$ is significant.

(e)
```
fit_3 <- lm(y ~ x2)
summary(fit_3)
```

```
##
## 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 0.0000137 ***
## ---
## 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: 0.00001366
```

The p-value associated with x_2 is significant and this model explains 17.63% of the variability of y. Both $\hat{\beta}_0$ and $\hat{\beta}_2$ are greater than β_0 and β_2, respectively.

We can reject $H_0 : \beta_2 = 0$, since the p-value associated with $\hat{\beta}_2$ is significant.

(f) Yes, the results in (c)-(e) contradict each other. In (c), we saw that we can't reject the null hypothesis $H_0 : \beta_2 = 0$, but in (e) we saw that we can. This is because x_1 and x_2 are highly correlated variables, with correlation > 0.8. Because of this, the presence of x_1 is masking the effect of x_2 and hence in (c) we saw that $\hat{\beta}_1$ is significant and $\hat{\beta}_2$ is not. When we removed x_1 from

the model in (e), since the highly correlated variable x_1 is removed, we saw that β_2 is significant too.

(g)

```
  x1 <- c(x1, 0.1)
  x2 <- c(x2, 0.8)
  y <- c(y, 6)

  # Full model
  fit_g_1 <- lm(y ~ x1 + x2)
  summary(fit_g_1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188,Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 0.000005564
```

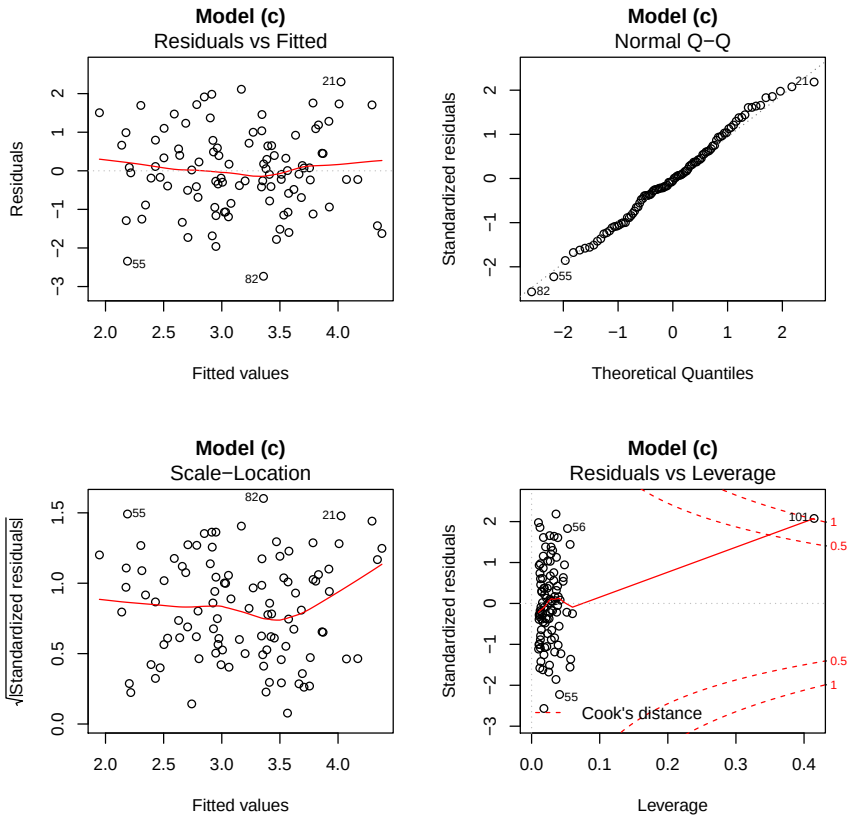
```
  # x_1 model
  fit_g_2 <- lm(y ~ x1)
  summary(fit_g_2)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562,Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 0.00004295
```

```
  # x_2 model
  fit_g_3 <- lm(y ~ x2)
  summary(fit_g_3)
```

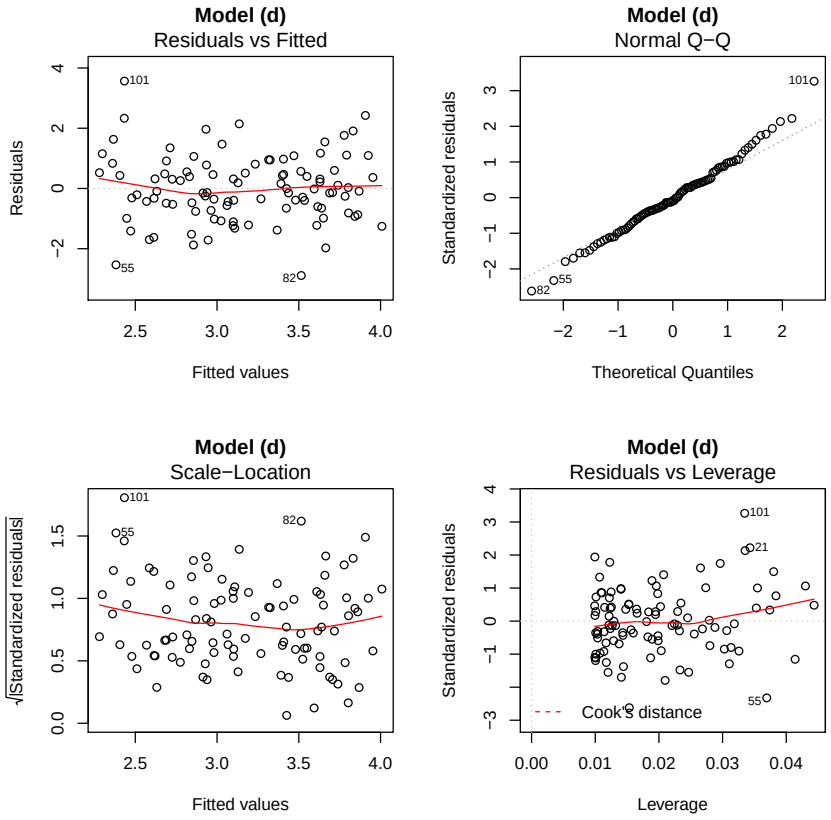
```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264    < 2e-16 ***
## x2            3.1190     0.6040   5.164 0.00000125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122,Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 0.000001253
```

```
par(mfrow=c(2,2))
plot(fit_g_1, main="Model (c)")
```



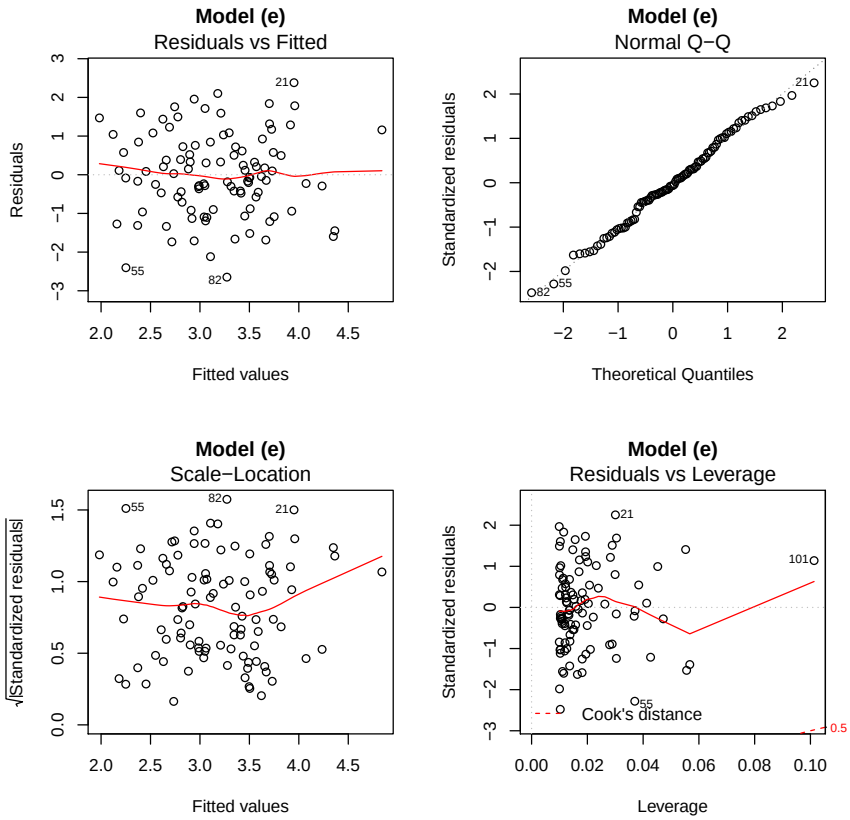
In model (c), adding the new observation improves the R^2 statistic and now, we can't reject the null hypothesis $H_0 : \beta_1 = 0$, but we can reject $H_0 : \beta_2 = 0$. Observation 101 is a high leverage point in this model and not an outlier.

```
par(mfrow=c(2,2))
plot(fit_g_2, main="Model (d)")
```



In model (d), adding the new observation reduced the R^2 statistic quite considerably. While observation 101 is an outlier, it is not a leveage point.

```
par(mfrow=c(2,2))
plot(fit_g_3, main="Model (e)")
```



In model (e), adding the new observation significantly improves the R^2 statistic and observation 101 is a high leverage point, though not an outlier.

Problem 11 (5 points)

Let $x_1, \ldots, x_n$ be a fixed set of input points and $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \overset{iid}{\sim} P_\epsilon$ with $\mathbb{E}(\epsilon_i) = 0$ and $\text{Var}(\epsilon_i) < \infty$. Prove that the MSE of a regression estimate $\hat{f}$ fit to $(x_1, y_1), \ldots, (x_n, y_n)$ for a random test point x_0 or $\mathbb{E}\left(y_0 - \hat{f}(x_0)\right)^2$ decomposes into variance, square bias, and irreducible error components. *Hint: You can apply the bias-variance decomposition proved in class.*

We apply the tower property, i.e. $\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X \mid Y))$, noting that we are conditioning on x_0.

The MSE at x_0 random is

$$\mathbb{E}\left((y_0 - \hat{f}(x_0))^2\right) = \mathbb{E}\left(\mathbb{E}\left((y_0 - \hat{f}(x_0))^2 \mid x_0\right)\right).$$

Using the decomposition in our lecture for fixed x_0:

$$\mathbb{E}\left((y_0 - \hat{f}(x_0))^2 \mid x_0\right) = \text{Var}\left(\hat{f}(x_0) \mid x_0\right) + \text{Bias}(\hat{f}(x_0) \mid x_0)^2 + \text{Var}(\epsilon)$$

Problem 12 (5 points)

Consider the regression through the origin model (i.e. with no intercept):

$$y_i = \beta x_i + \epsilon_i \tag{3}$$

(a) *(1 point)* Find the least squares estimate for β.

To find the least squares estimate for β, we want to minimize the residual sum of squares, i.e.

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

$$= \sum_{i=1}^{n}(y_i - \beta x_i)^2 \tag{5}$$

$$\tag{6}$$

By taking the first derivative and setting it equal to zero, we get

$$\frac{\partial RSS}{\partial \beta} = -2\sum_{i=1} n x_i(y_i - \beta x_i) = 0 \tag{7}$$

$$\Rightarrow \quad \hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2} \tag{8}$$

The second derivative tells us whether the RSS is concave or convex.

$$\frac{\partial^2 RSS}{\partial \beta^2} = 2\sum_{i=1}^{n} x_i^2 \geq 0 \tag{9}$$

which confirms that the function is convex and that $\hat{\beta}$ is the least squares estimate for β.

(b) *(2 points)* Assume $\epsilon_i \overset{iid}{\sim} P_\epsilon$ such that $\mathbb{E}(\epsilon_i) = 0$ and $\text{Var}(\epsilon_i) = \sigma^2 < \infty$. Find the standard error of the estimate.

The variance of $\hat{\beta}$ is

$$\text{Var}\left(\hat{\beta}\right) = var\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}\right) \tag{10}$$

$$= \frac{\sum_{i=1}^{n} x_i^2 \text{Var}(y_i)}{(\sum_{i=1}^{n} x_i^2)^2} \tag{11}$$

$$= \frac{\sum_{i=1}^{n} x_i^2 \text{Var}(\epsilon_i)}{(\sum_{i=1}^{n} x_i^2)^2} \tag{12}$$

Assuming that $\epsilon_i \overset{iid}{\sim} P_\epsilon$ such that $\mathbb{E}(\epsilon_i) = 0$ and $\text{Var}(\epsilon_i) = \sigma^2 < \infty$, then

$$\text{Var}\left(\hat{\beta}\right) = \frac{\sum_{i=1}^{n} x_i^2 \sigma^2}{(\sum_{i=1}^{n} x_i^2)^2} = \frac{\sigma^2}{\sum_{i=1}^{n} x_i^2} \tag{13}$$

The standard error is therefore

$$se(\hat{\beta}) = \sqrt{\text{Var}\left(\hat{\beta}\right)} = \frac{\sigma}{\sqrt{\sum_{i=1}^{n} x_i^2}} \tag{14}$$

(c) *(2 points)* Find conditions that guarantee that the estimator is consistent. *n.b. An estimator $\hat{\beta}_n$ of a parameter β is consistent if $\hat{\beta} \xrightarrow{p} \beta$, i.e. if the estimator converges to the parameter value in probability.*

Again assuming that $\epsilon_i \overset{iid}{\sim} P_\epsilon$ such that $\mathbb{E}(\epsilon_i) = 0$ and $\text{Var}(\epsilon_i) = \sigma^2 < \infty$, then

$$\mathbb{E}\left(\hat{\beta}\right) = \mathbb{E}\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}\right) \tag{15}$$

$$= \frac{\sum_{i=1}^{n} x_i \mathbb{E}(y_i)}{\sum_{i=1}^{n} x_i^2} \tag{16}$$

$$= \frac{\sum_{i=1}^{n} x_i^2 \beta}{\sum_{i=1}^{n} x_i^2} \tag{17}$$

$$= \beta \tag{18}$$

Since $\hat{\beta}_n$ is an unbiased estimator, $\mathbb{E}\left(\hat{\beta}-\beta\right)^2 = \text{Var}\left(\hat{\beta}\right) = \frac{\sigma^2}{\sum_{i=1}^n x_i^2}$. If there are an infinite number of non-zero x_i's such that as $n \to \infty$, $\sum_{i=1}^n x_i^2 \to \infty$, then $\mathbb{E}\left(\hat{\beta}-\beta\right)^2 \to 0$; i.e. if the sequence of x_i's has the aforementioned property, then $\hat{\beta}_n$ converges to β in quadratic mean. Since convergence in quadratic mean is a stronger condition than convergence in probability, the aforementioned property guarantees that $\hat{\beta}_n$ is a consistent estimator.

Aside. We can alternatively use the Chebyshev's inequality, which tells us that $\hat{\beta}_n$ is a consistent estimator of β if and only if

$$P(|\hat{\beta}-\beta| > \delta) \to 0 \; \forall \delta > 0. \tag{19}$$

Since

$$\begin{aligned}
\hat{\beta}-\beta &= \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} - \beta && (20)\\
&= \frac{\sum_{i=1}^n x_i(\beta x_i + \epsilon_i)}{\sum_{i=1}^n x_i^2} - \beta && (21)\\
&= \frac{\sum_{i=1}^n x_i \epsilon_i}{\sum_{i=1}^n x_i^2} && (22)
\end{aligned}$$

By Chebyshev's inequality,

$$\begin{aligned}
P\left(|\hat{\beta}-\beta| > \delta\right) &= P\left(\left|\frac{\sum_{i=1}^n x_i \epsilon_i}{\sum_{i=1}^n x_i^2}\right| > \delta\right) && (23)\\
&\leq \frac{\text{Var}\left(\frac{\sum_{i=1}^n x_i \epsilon_i}{\sum_{i=1}^n x_i^2}\right)}{\delta^2} && (24)\\
&= \frac{\sigma^2}{\delta^2 \sum_{i=1}^n x_i^2} && (25)
\end{aligned}$$

Thus, if $\frac{\sigma^2}{\delta^2 \sum_{i=1}^n x_i^2} \overset{n\to\infty}{\longrightarrow} 0$, then $P(|\hat{\beta}-\beta| > \delta) \to 0$. That is, if there are an infinite number of non-zero x_i's such that as $n \to \infty$, $\sum_{i=1}^n x_i^2 \to \infty$, then $\hat{\beta}$ will be a consistent estimator.