Homework 5 - Stats 230 (chapter 4.1, 4.2 and 4.4)

Dhyey Mavani

date

MLR III - Added variable plots, Unusual points and Variable Selection Techniques (High Peaks & MLB wins)

PROBLEMS TO TURN IN: #4.2, #4.3, #4.12

Note: A lot of code is provided for this assignment, but you'll still need to fit models and do some computations.

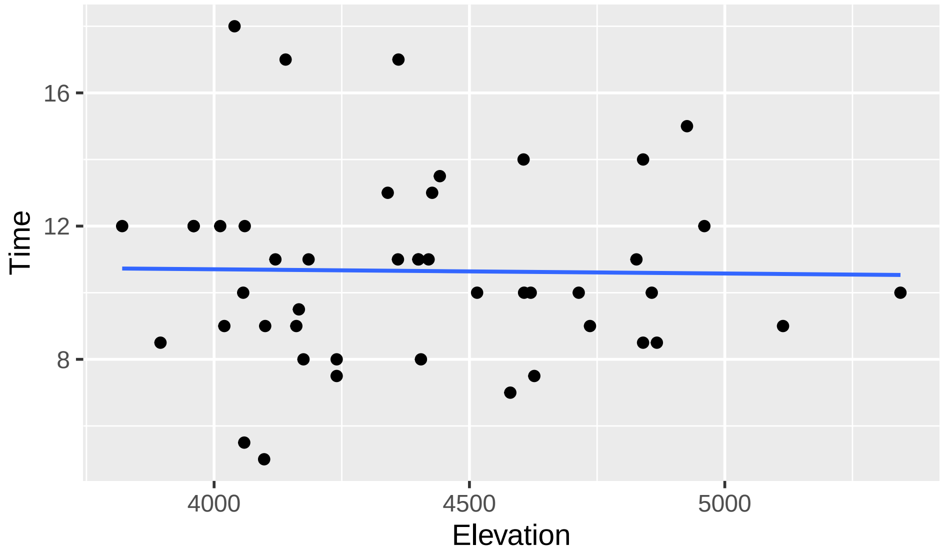
```
data(HighPeaks)
```

Exercise 4.2

4.2 part a:

SOLUTION: We can see that the correlation between Time and Elevation is slightly negative and I don't think its looks like that Elevation should be very helpful in predicting Time because the correlation coefficient is very small (around -0.01 in this case).

```
gf_point(Time ~ Elevation, data = HighPeaks) %>%
  gf_lm()
```



```
cor(Time ~ Elevation, data = HighPeaks)
```

```
## [1] -0.0162768
```

4.2 part b: (Has 2 main components to it, requiring you to fit 3 models)

SOLUTION:

We can see that model2 performs much better than model1 as the Rˆ2 value is significantly higher. But, we can also see that when we have a model with just Length compared to two-predictor model with Length and Elevation, we don't have a significant difference in Rˆ2 value which tells us that this two-predictor model does significantly better at explaining Time compared to the model with Elevation alone, but not compared to the model with Length alone. But since Elevation is still significant according to the p-value in our two-predictor model, I would prefer two-predictor model even though it just does marginally better than the model with just Length.

```
model1 <- lm(Time ~ Elevation, data = HighPeaks)
msummary(model1)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.211376   5.195380    2.16    0.036 *
## Elevation   -0.000127   0.001176   -0.11    0.915
##
## Residual standard error: 2.83 on 44 degrees of freedom
## Multiple R-squared:  0.000265,   Adjusted R-squared:  -0.0225
## F-statistic: 0.0117 on 1 and 44 DF,  p-value: 0.915
```

```
model2 <- lm(Time ~ Elevation + Length, data = HighPeaks)
msummary(model2)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.075379   2.532713    3.19   0.0027 **
## Elevation   -0.001448   0.000581   -2.49   0.0165 *
## Length       0.712334   0.059333   12.01  2.5e-15 ***
##
## Residual standard error: 1.37 on 43 degrees of freedom
## Multiple R-squared:  0.77,   Adjusted R-squared:  0.76
## F-statistic: 72.1 on 2 and 43 DF,  p-value: 1.84e-14
```

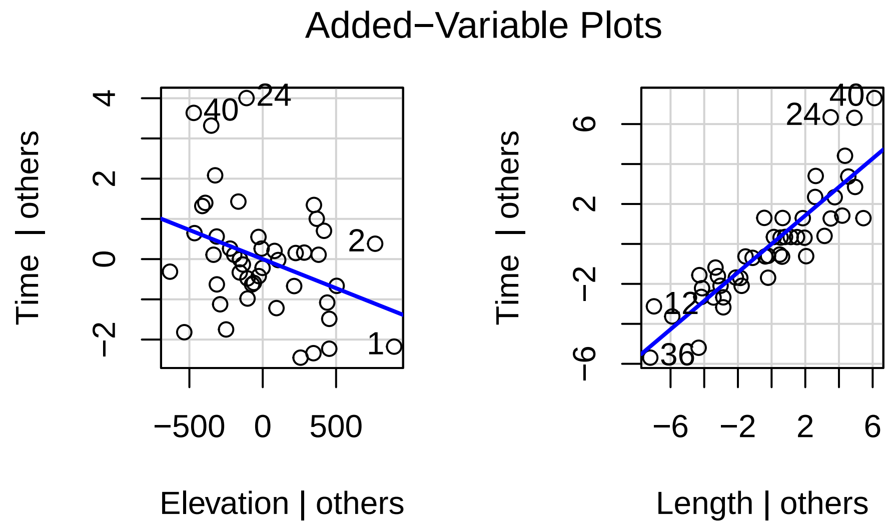
```
model3 <- lm(Time ~ Length, data = HighPeaks)
msummary(model3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0482     0.8037    2.55    0.014 *
## Length        0.6843     0.0616   11.11  2.4e-14 ***
##
## Residual standard error: 1.45 on 44 degrees of freedom
## Multiple R-squared:  0.737,  Adjusted R-squared:  0.731
## F-statistic:  123 on 1 and 44 DF,  p-value: 2.39e-14
```

4.2 part c:

SOLUTION: We can see that the slopes of the added-variable plots are significantly negative and positive in the case of Elevation and Length respectively, hence it would make sense for us to keep both variables in our model of choice. Given this, we see a few points that may be unusual in both of these plots, but in my opinion the association is moderately strong.

```
car::avPlots(model2)
```



Exercise 4.3 Note: Allowed predictors are all variables in the data set after WinPct. (It says any predictors after Wins and Losses, but Team is an identifier and League is not included in the text solution, so leave it out).

```
data("MLBStandings2016")
Standings <- MLBStandings2016 #renamed for faster typing
```

4.3 part a:

SOLUTION: The predictors in our 4-predictor model are Runs, ERA, Saves and WHIP. The R^2 for this model is 88.6%

```
#this will run the forward regression for you, the problem asks you to do more with the output.
forward <- regsubsets(WinPct ~ BattingAverage + Runs + Hits + HR + Doubles + Triples
                      + RBI + SB + OBP + SLG + ERA + HitsAllowed + Walks + StrikeOuts
                      + Saves + WHIP, data = Standings, method = "forward", nbest = 1)
with(summary(forward), data.frame(rsq, cp, outmat))
```

```
##               rsq       cp BattingAverage Runs Hits HR Doubles Triples RBI SB
## 1  ( 1 ) 0.636494 73.58815
## 2  ( 1 ) 0.810810 27.83161                   *
## 3  ( 1 ) 0.866483 14.57898                   *
## 4  ( 1 ) 0.886258 11.16134                   *
## 5  ( 1 ) 0.897813  9.99583                   *               *
## 6  ( 1 ) 0.911756  8.17573              *    *               *
## 7  ( 1 ) 0.917953  8.47805              *    *               *
## 8  ( 1 ) 0.923515  8.95413              *    *               *
##          OBP SLG ERA HitsAllowed Walks StrikeOuts Saves WHIP
## 1  ( 1 )           *
## 2  ( 1 )           *
## 3  ( 1 )           *                                  *
## 4  ( 1 )           *                                  *    *
## 5  ( 1 )           *                                  *    *
## 6  ( 1 )           *                                  *    *
## 7  ( 1 )   *       *                                  *    *
## 8  ( 1 )   *       *                 *                *    *
```

```
model <- lm(WinPct ~ Runs + ERA + Saves + WHIP, data = Standings)
msummary(model)
```

```
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  5.16e-01   1.19e-01    4.35  0.00020 ***
## Runs         5.19e-04   7.76e-05    6.68  5.3e-07 ***
## ERA         -3.64e-02   2.63e-02   -1.38  0.17840    
## Saves        2.64e-03   6.79e-04    3.89  0.00065 ***
## WHIP        -2.66e-01   1.27e-01   -2.08  0.04746 *  
## 
## Residual standard error: 0.024 on 25 degrees of freedom
## Multiple R-squared:  0.886,  Adjusted R-squared:  0.868 
## F-statistic: 48.7 on 4 and 25 DF,  p-value: 1.91e-11
```

4.3 part b:

SOLUTION: The predictors in our 4-predictor model are BattingAverage, Runs, Saves and WHIP. The R^2 for this model is 88.4%

```
backward <- regsubsets(WinPct ~ BattingAverage + Runs + Hits + HR + Doubles +
                      Triples + RBI + SB + OBP + SLG + ERA + HitsAllowed + Walks
                      + StrikeOuts + Saves + WHIP, data = Standings,
                      method = "backward", nbest = 1)
with(summary(backward), data.frame(rsq, cp, outmat))
```

```
##               rsq       cp BattingAverage Runs Hits HR Doubles Triples RBI SB
## 1  ( 1 ) 0.605598 82.05260
## 2  ( 1 ) 0.765323 40.29330                   *
## 3  ( 1 ) 0.877536 11.55102                   *
## 4  ( 1 ) 0.883617 11.88500              *    *
## 5  ( 1 ) 0.899731  9.47024              *    *    *
## 6  ( 1 ) 0.905400  9.91721              *    *    *
## 7  ( 1 ) 0.915554  9.13543              *    *    *
## 8  ( 1 ) 0.935313  5.72189              *    *    *
##          OBP SLG ERA HitsAllowed Walks StrikeOuts Saves WHIP
## 1  ( 1 )                                                   *
## 2  ( 1 )                                                   *
## 3  ( 1 )                                              *    *
## 4  ( 1 )                                              *    *
## 5  ( 1 )                                              *    *
## 6  ( 1 )                            *                 *    *
## 7  ( 1 )                      *     *                 *    *
## 8  ( 1 )   *                  *     *                 *    *
```

```
model <- lm(WinPct ~ BattingAverage + Runs + Saves + WHIP, data = Standings)
msummary(model)
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.464149   0.140194    3.31  0.00283 **
## BattingAverage  0.781844   0.684072    1.14  0.26390
## Runs            0.000427   0.000116    3.67  0.00114 **
## Saves           0.002861   0.000641    4.46  0.00015 ***
## WHIP           -0.448925   0.060373   -7.44  8.7e-08 ***
##
## Residual standard error: 0.0243 on 25 degrees of freedom
## Multiple R-squared:  0.884,  Adjusted R-squared:  0.865
## F-statistic: 47.5 on 4 and 25 DF,  p-value: 2.54e-11
```

4.3 part c:

SOLUTION: The predictors in our 4-predictor model are Runs, Doubles, Saves and WHIP. The R^2 for this model is 88.9%

```
best <- regsubsets(WinPct ~ BattingAverage + Runs + Hits + HR + Doubles + Triples
                   + RBI + SB + OBP + SLG + ERA + HitsAllowed + Walks + StrikeOuts
                   + Saves + WHIP, data = Standings, nbest = 1)
with(summary(best), data.frame(rsq, cp, outmat))
```

```
##                rsq       cp BattingAverage Runs Hits HR Doubles Triples RBI SB
## 1  ( 1 ) 0.636494 73.58815
## 2  ( 1 ) 0.810810 27.83161                     *
## 3  ( 1 ) 0.877536 11.55102                     *
## 4  ( 1 ) 0.888539 10.53656                     *                *
## 5  ( 1 ) 0.901883  8.88069                *    *                *
## 6  ( 1 ) 0.911899  8.13665                *    *                *
## 7  ( 1 ) 0.918283  8.38757                *         *  *
## 8  ( 1 ) 0.935313  5.72189                *    *    *
##          OBP SLG ERA HitsAllowed Walks StrikeOuts Saves WHIP
## 1  ( 1 )           *
## 2  ( 1 )           *
## 3  ( 1 )                                              *    *
## 4  ( 1 )                                              *    *
## 5  ( 1 )                                              *    *
## 6  ( 1 )                            *                 *    *
## 7  ( 1 )                      *     *                 *    *
## 8  ( 1 )   *                  *     *                 *    *
```

```
model <- lm(WinPct ~ Runs + Doubles + Saves + WHIP, data = Standings)
msummary(model)
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.621685   0.122659    5.07  3.1e-05 ***
## Runs          0.000635   0.000104    6.11  2.2e-06 ***
## Doubles      -0.000446   0.000284   -1.57   0.1288
## Saves         0.002527   0.000691    3.66   0.0012 **
## WHIP         -0.427702   0.055296   -7.73  4.3e-08 ***
##
## Residual standard error: 0.0238 on 25 degrees of freedom
## Multiple R-squared:  0.889,  Adjusted R-squared:  0.871
## F-statistic: 49.8 on 4 and 25 DF,  p-value: 1.49e-11
```

4.3 part d:

SOLUTION: Mallow's Cps for four predictor models in parts (a), (b) and (c) are 11.16134, 11.88500 and 10.53656 respectively.

4.3 part e:

SOLUTION: We can see that the four predictor model from part (c) has the lowest Cp and highest R^2 among others.

Exercise 4.12

4.12 part a:

SOLUTION: I think the four predictor model would be my choice since it has highest R^2 and lowest Cp. Going with 3 predictor model doesn't make sense for me because I think it is worth it to add an additional predictor which lowers Cp by over 1, while increasing R^2 and only slightly increasing the complexity of interpreting the model.

```
best <- regsubsets(Time ~ Elevation + Difficulty + Ascent + Length, data = HighPeaks, nbest = 1)
with(summary(best), data.frame(rsq, cp, outmat))
```

```
##                rsq      cp Elevation Difficulty Ascent Length
## 1  ( 1 ) 0.737036 25.4122                                   *
## 2  ( 1 ) 0.796218 12.2405                      *            *
## 3  ( 1 ) 0.827202  6.2977         *            *            *
## 4  ( 1 ) 0.840066  5.0000         *            *      *     *
```

4.12 part b:

SOLUTION:

We can see that there is a linear relationship.

Equal (Constant) Variance - Assess with a residual vs. fitted plot. Looking for no pattern Next, we can check the condition for constant variance of errors by looking at the residual vs fitted plots. A few outliers are noted in red but nothing exerting a great deal of leverage (unduly influencing the overall relationship). There is no pattern of errors (looks like a cloud) and is mostly linear. IF the variance in Y was not equal across X we could see a fan shape indicating heteroscedastcity which would need to be resolved with a transformation.

Normality - Assess the distribution of errors with histograms/qqplots of residuals errors should be centered at zero (ZERO MEAN), no skew or pattern (RANDOM)- necessary for inference. The histogram is relatively normal and centered at zero (latter condition is always true for least squares technique). There are a few high residuals > 15 that many need to be considered. All other residuals fall between +/- 15. In the QQPlot, we can see that most points are on the line, there is no real shape and only a few are off at the ends, hence we are good there!

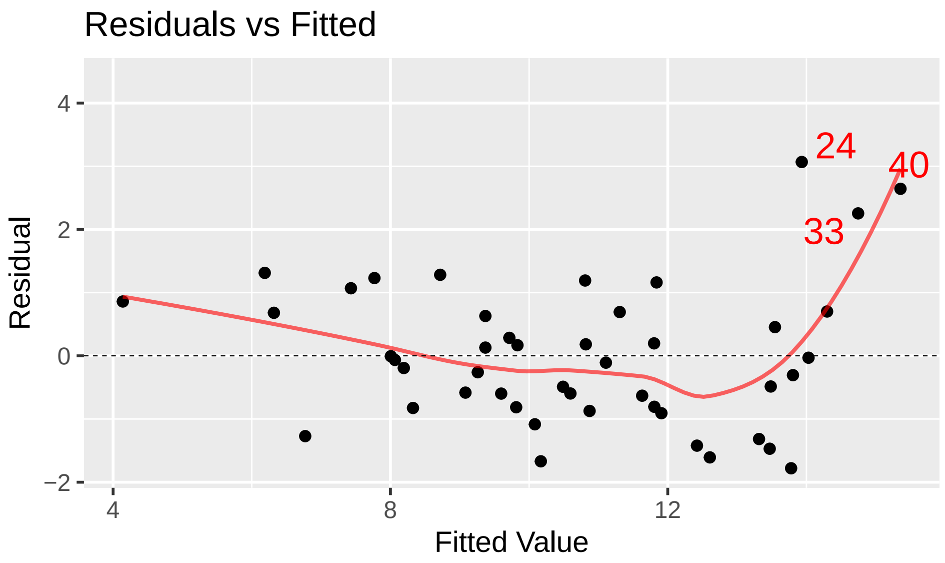
Thus, I can say that the conditions look fine for the reasonability of the model.

```
model <- lm(Time ~ Elevation + Difficulty + Ascent + Length, data = HighPeaks)
msummary(model)
```

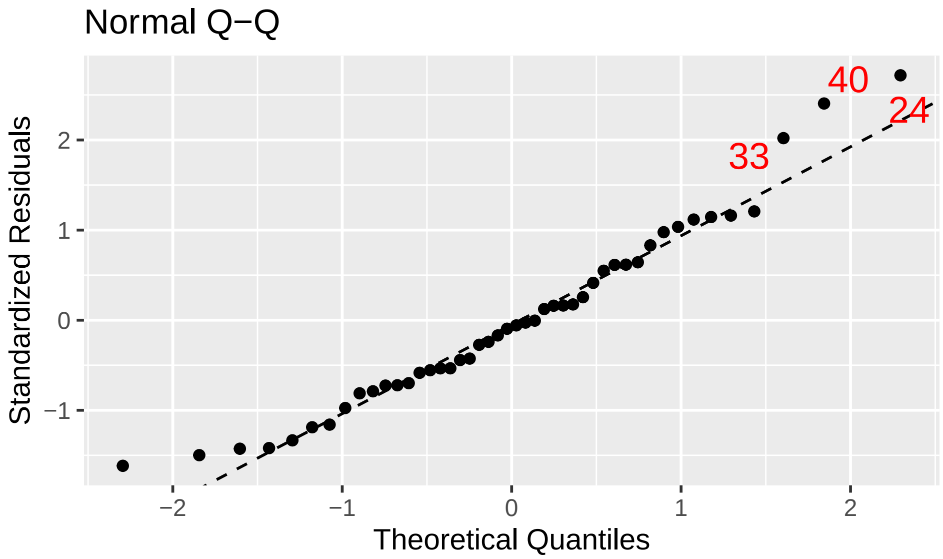
```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.956786   2.230763    2.67  0.01082 *
## Elevation    -0.001670   0.000518   -3.22  0.00249 **
## Difficulty    0.865453   0.228528    3.79  0.00049 ***
## Ascent        0.000601   0.000331    1.82  0.07669 .
## Length        0.444008   0.081252    5.46  2.5e-06 ***
##
## Residual standard error: 1.17 on 41 degrees of freedom
## Multiple R-squared:  0.84,   Adjusted R-squared:  0.824
## F-statistic: 53.8 on 4 and 41 DF,  p-value: 8.74e-16
```

```
mplot(model, which = 1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
mplot(model, which = 2)
```



4.12 part c:

SOLUTION: The unusual mountains are Seward Mtn., Mt. Donaldson, and Mt. Emmons with studentised residuals of 2.96456, 2.10301 and 2.56285 respectively. This is because their studentized residuals are greater than 2.

```
HighPeaksAug <- augment(model) %>% mutate(.stu.resid = rstudent(model))
filteredHighPeaksAug <- HighPeaksAug %>% filter(abs(.stu.resid) >= 2)
filteredHighPeaksAug
```

```
## # A tibble: 3 x 12
##    Time Elevation Difficulty Ascent Length .fitted .resid   .hat .sigma .cooksd
##   <dbl>     <int>      <int>  <int>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
## 1    17      4361          7   3490     16    13.9   3.07 0.0707   1.07  0.112
## 2    17      4140          7   3490     17    14.7   2.25 0.0919   1.12  0.0826
## 3    18      4040          7   3490     18    15.4   2.64 0.119    1.10  0.156
## # ... with 2 more variables: .std.resid <dbl>, .stu.resid <dbl>
```

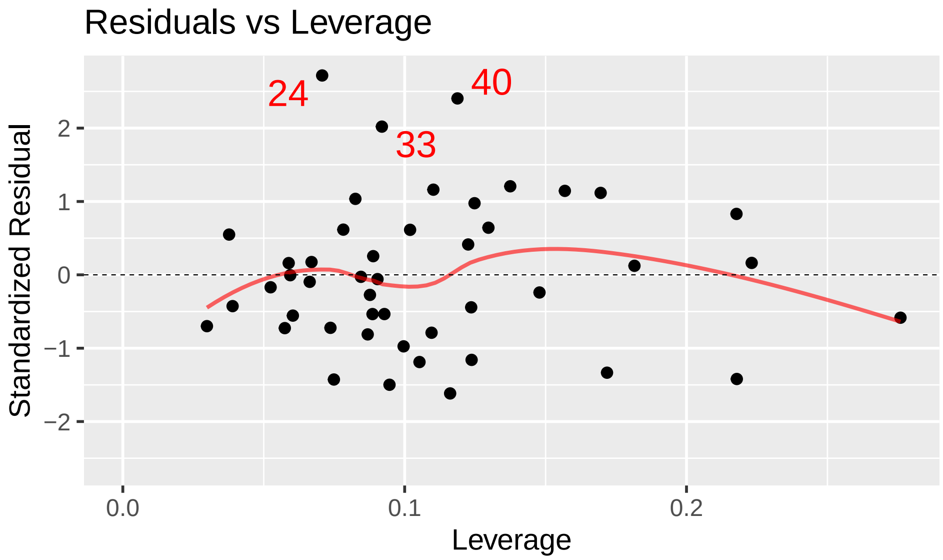
```
#filter for unusual values; abs() in R is absolute value
```

4.12 part d:

SOLUTION: We know that Leverage cutoff in this case is 2(4+1)/46 = 0.217391. Also, the Cook's distance cutoff is 0.5. Now we can continue our analysis. We see that the Cook's distance stays less than 0.5 for all observations, hence we don't have to worry about that unusualness. But, we can see that the mountains Mt. Marcy, Cascade Mtn., Cliff Mtn. and Nye Mtn. have hat values of 0.223127, 0.217738, 0.217848 and 0.275927 respectively which are greater than the moderate leverage cutoff, which implies that these data points are unusual in the sense that they have high leverage.

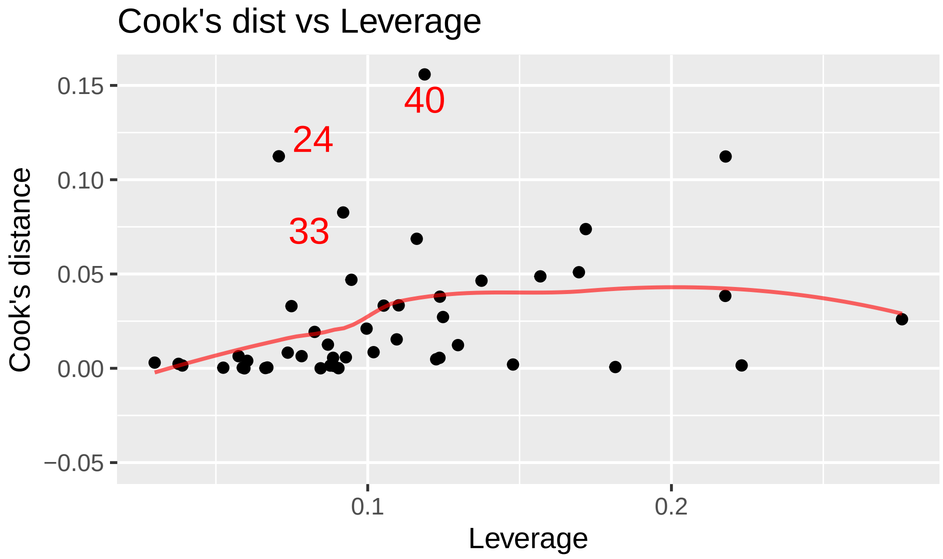
```
#suggest using plots and numeric values to support your response
#_values are saved in HighPeaksAug
mplot(model, which = 5)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
mplot(model, which = 6)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
filterHighPeaksAug <- HighPeaksAug %>% filter(abs(.hat) >= 0.217391)
filterHighPeaksAug
```

```
## # A tibble: 4 x 12
##    Time Elevation Difficulty Ascent Length .fitted .resid  .hat .sigma .cooksd
##   <dbl>     <int>      <int>  <int>  <dbl>   <dbl>  <dbl> <dbl>  <dbl>   <dbl>
## 1  10        5344          5   3166   14.8    9.83  0.168 0.223   1.19 0.00152
## 2   5        4098          2   1940    4.8    4.14  0.860 0.218   1.18 0.0384
## 3  12        3960          6   2160   17.2   13.5  -1.47  0.218   1.16 0.112
## 4   8.5      3895          6   1844    7.5    9.08 -0.582 0.276   1.18 0.0260
## # ... with 2 more variables: .std.resid <dbl>, .stu.resid <dbl>
```

```
#
```