Lab 06

Rohit Kulkarni

02/22/2022

Data

The dataset in this lab contains the SAT score (out of 1600) and other variables that may be associated with SAT performance for each of the 50 states in the U.S. The data are based on test takers for the 1982 exam. The following variables are in the dataset:

- SAT: average total SAT score
- State: U.S. State
- Takers: percentage of high school seniors who took exam
- Income: median income of families of test-takers ($ hundreds)
- Years: average number of years test-takers had formal education in social sciences, natural sciences, and humanities
- Public: percentage of test-takers who attended public high schools
- Expend: total state expenditure on high schools ($ hundreds per student)
- Rank: median percentile rank of test-takers within their high school classes

This is the same dataset we used in class on February 8th.

Exercises

##Part 1: Model Selection We begin this lab by conducting model selection with various selection criteria to choose a final model from the SAT dataset. The code to load the data and create the full main effects model is shown below. The next few questions will walk you through backward model selection using different model selection criteria to select a model.

```
sat_scores <- Sleuth3::case1201
full_model <- lm(SAT ~ Takers + Income + Years + Public + Expend + Rank , data = sat_scores)
tidy(full_model)
```

Type `??regsubsets` in the console for more information about the regsubsets function.‘

1. We will use the `regsubsets` function in the leaps R package to perform backward selection on multiple linear regression models with $Adj.R^2$ or BIC as the selection criteria.

Fill in the code to display the model selected from backward selection with $Adj.R^2$ as the selection criterion.

```
model_select <- regsubsets(SAT ~ Takers + Income + Years + Public + Expend +
                                Rank , data = sat_scores, method = "backward")
select_summary <- summary(model_select)
select_summary$adjr2 #display coefficients
```

2. Fill in the code below to display the model selected from backward selection with BIC as the selection criterion.

```
select$summary$bic #display coefficients
```

Type ??step in the console for more information about the step function. The output from the step function will show you the output from each step of the selection phase.

Next, let's select a model using AIC as the selection criterion. To select a model using AIC, we will use the step function in R. The code below is to conduct backward selection using AIC as the criterion and store the selected model in an object called model_select_aic. Use the tidy function to display the coefficients of the selected model.

```
model_select_aic <- step(full_model, direction = "backward")
tidy(model_select_aic)
```

4. Compare the final models selected by Adj.R2, AIC, and BIC. *Do the models have the same number of predictors?* If they don't have the same number of predictors, which selection criterion resulted in the model with the fewest number of predictors? Is this what you would expect? Briefly explain.

Model selected by Adj.R2:

```
select_summary$adjr2
```

0.7695367 0.8405479 0.8627047 0.8661268 0.8649009 0.8617684

Model selected by BIC:

```
select_summary$bic
```

-66.59010 -82.14815 -86.79191 -85.24089 -81.99674 -78.08808

Model selected by AIC:

```
model_select_aic <- step(full_model, direction = "backward")
tidy(model_select_aic)
```

```
1 (Intercept) -205.      118.        -1.74 8.90e- 2
2 Years         21.9       6.04       3.63 7.31e- 4
3 Public        -0.664     0.450     -1.48 1.47e- 1
4 Expend         2.24      0.678      3.31 1.87e- 3
5 Rank          10.0       0.603     16.6  8.67e-21
```

The model selected by AIC only has five predictors, while the other models have 6. This makes sense because the BIC model favors a higher number of predictors, whether or not those predictors are actually useful in creating an accurate model.

##Part II: Model Diagnostics Let's choose model_select_aic, the model selected usng AIC, to be our final model. In this part of the lab, we will examine some model diagnostics for this model.

5. Use the augment function to create a data frame that contains model predictiosn and statistics for each observation. Save the data frame, and add a variable called obs_num that contains the observation (row) number. Display the first 5 rows of the new data frame.

```
sat_scores <- augment(model_select_aic, sat_scores)
sat_scores$obs_num <- seq.int(nrow(sat_scores))
head(sat_scores, 5)
```

```
  State        SAT Takers Income Years Public Expend  Rank obs_num .fitted .resid   .hat .sigma .cooksd
  <fct>      <int>  <int>  <int> <dbl>  <dbl>  <dbl> <dbl>   <int>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
1 Iowa        1088      3    326  16.8   87.8   25.6  89.7       1   1059.  28.7  0.100    25.8 0.0304
2 SouthDako~  1075      2    264  16.1   86.2   20.0  90.6       2   1041.  34.0  0.0788   25.7 0.0320
3 NorthDako~  1068      3    317  16.6   88.3   20.6  89.8       3   1044.  24.0  0.0894   25.9 0.0185
4 Kansas      1045      5    338  16.3   83.9   27.1  86.3       4   1021.  24.4  0.0585   25.9 0.0117
5 Nebraska    1045      5    293  17.2   83.6   21.0  88.5       5   1050.  -4.99 0.113    26.2 0.00106
```

6. Let's examine the leverage for each observation. Based on the lecture notes, what threshold should we use to determine if observations in this dataset have high leverage? Report the value and show the quation you used to calculate it.

I will use the rule that a threshold for high leverage observations should be 3 times the number of parameters divided by the number of observations. In this dataset, that would be (3 * 4)/50 which is 0.24.

7. Plot the leverage (.hat) vs. the observation number. Add a line on the plot marking the threshold from the previous exercise. Be sure to include an informative title and clearly label the axes. You can use geom_hline to the add the threshold line to the plot.

```
p <- ggplot(sat_scores, aes(x=obs_num, y=.hat)) + geom_point() + geom_hline(yintercept=0.24) + xlab("Ob
  ylab("Observation Leverage") + ggtitle("Leverage of Observations")
p
```

8. Which states (if any) in the dataset are considered high leverage? Show the code used to determine the states. Hint: You may need to get State from sat_data.

```
high_leverage_states <- filter(sat_scores, .hat > 0.24)
head(high_leverage_states)
```

Using this code, I found that Louisiana and Alaska are states in the dataset with high leverage.

9. Next, we will examine the standardized residuals. Plot the standardized residuals (.std.resid) versus the predicted values. Include horizontal lines at y=2 and y= negative 2 indicating the thresholds used to determine if standardized residuals have a large magnitude. Be sure to include an informative title and clearly label the axes.You can use geom_hline to the add the threshold lines to the plot.

```
p <- ggplot(sat_scores, aes(x=.fitted, y=.std.resid)) + geom_point() + geom_hline(yintercept=2) + geom_
p
```

10. Based on our thresholds, which states (if any) are considered to have standardized residuals with large magnitude? Show the code used to determine the states. Hint: You may need to get State from sat_data.

```
out_of_bouds_std_resid <- filter(sat_scores, .std.resid < -2 | .std.resid > 2)
head(out_of_bouds_std_resid)
```

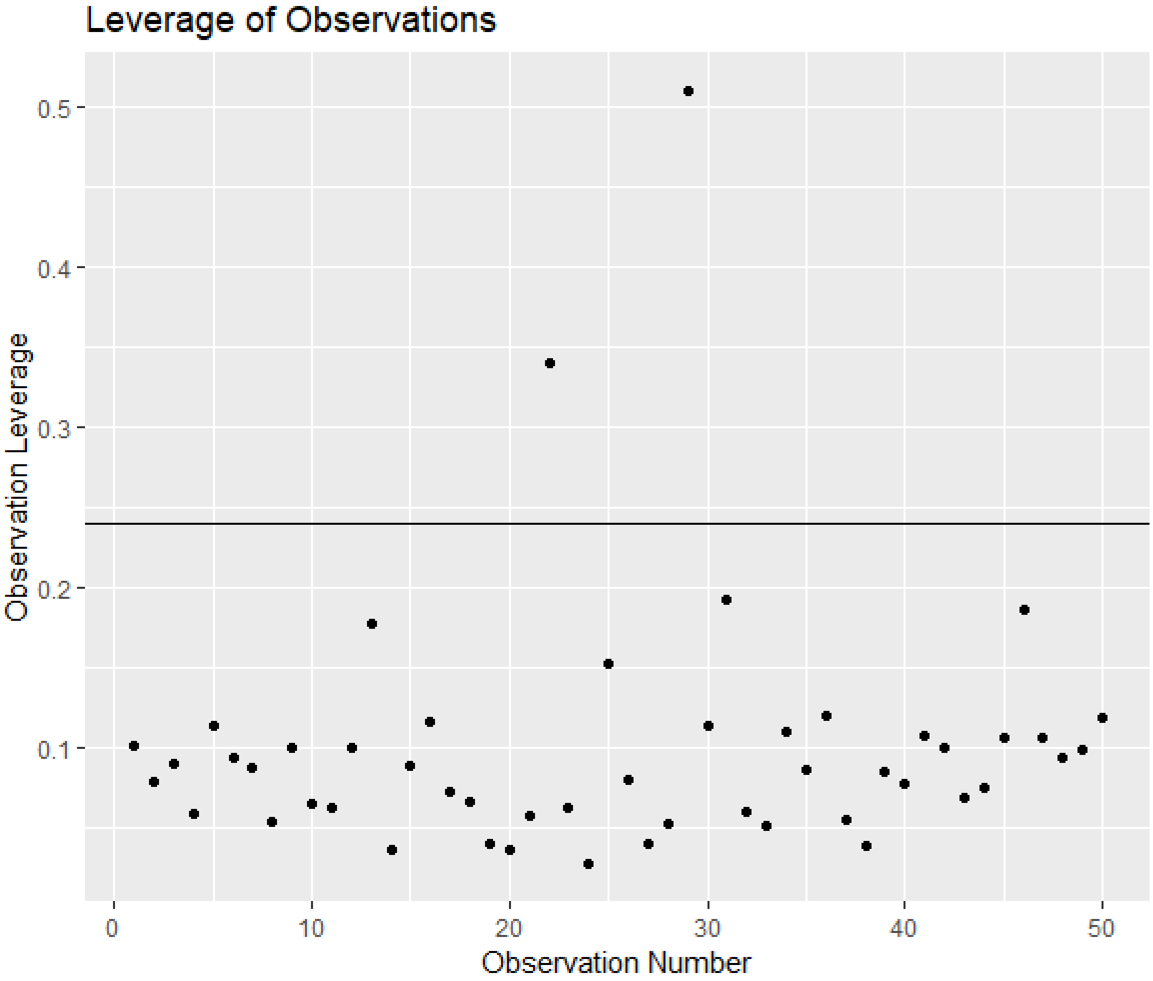

Figure 1: Leverage of Observations

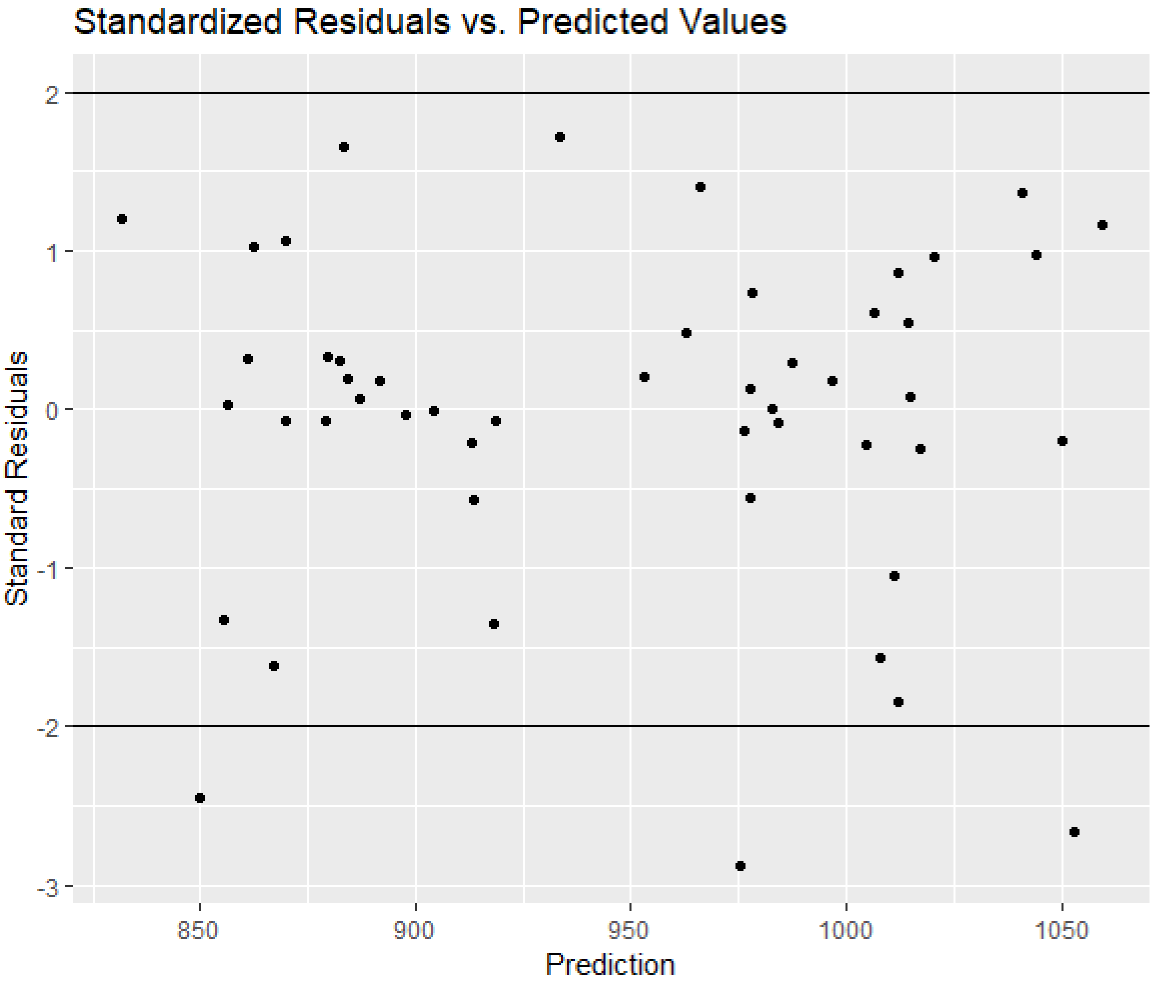

Figure 2: Predicted Values vs. Standardized Residuals

According to the code I used, Mississippi, Alaska, and South Carolina have standardized residuals with large magnitude.

11. Let's determine if any of these states with high leverage and/or high standardized residuals are influential points, i.e. are significantly impacting the coefficients of the model. Plot the Cook's Distance (.cooksd) vs. the observation number. Add a line on the plot marking the threshold to determine a point is influential. Be sure to include an informative title and clearly label the axes. You can use geom_hline to the add the threshold line to the plot.

```
p <- ggplot(sat_scores, aes(x=obs_num, y=.cooksd)) + geom_point() + geom_hline(yintercept=1) + xlab("Obs
p
```

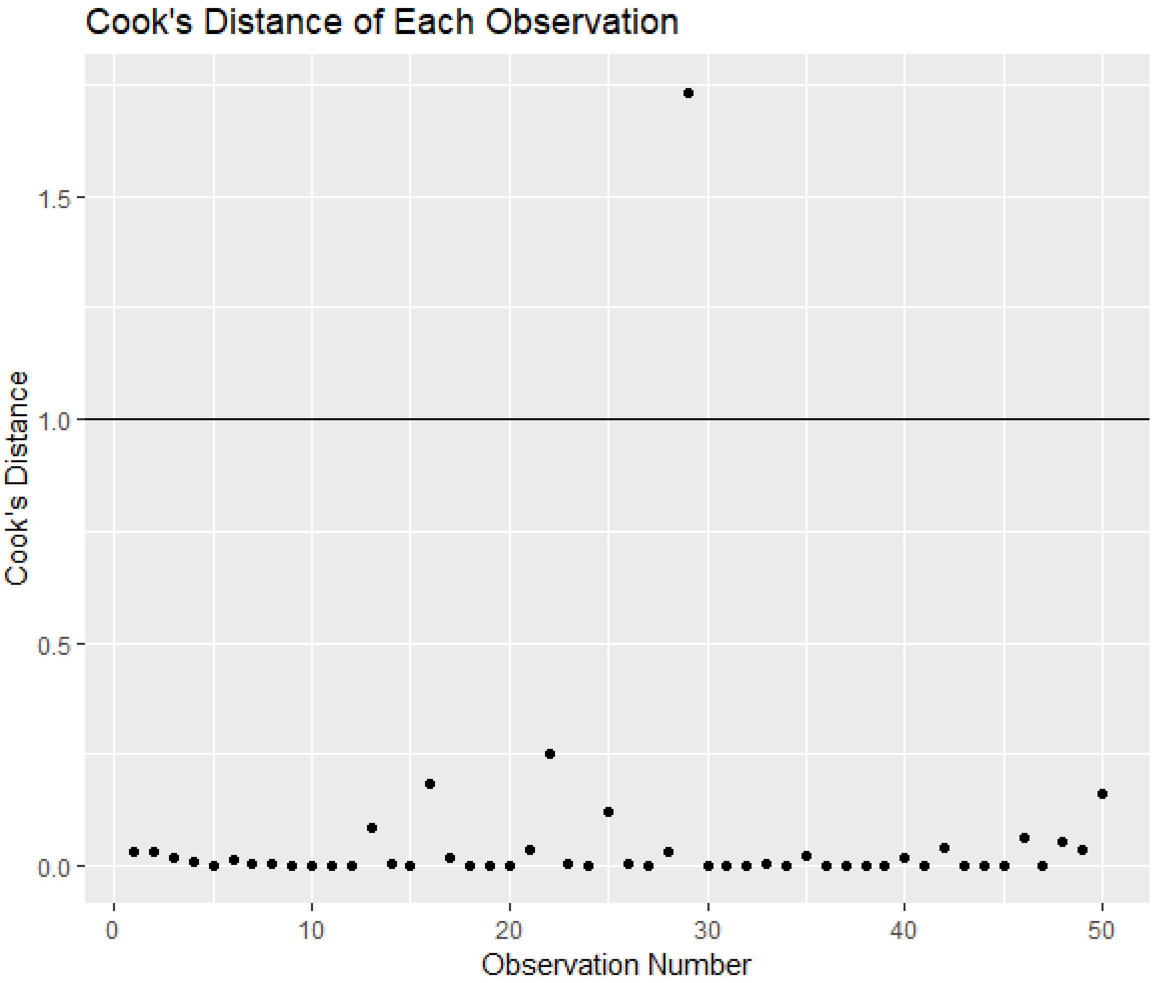

Figure 3: Cook's Distance of Each Observation

- Which states (if any) are considered to be influential points?

```
out_of_bounds_cooksd <- filter(sat_scores, .cooksd > 1)
head(out_of_bounds_cooksd)
```

According to this code, Alaska is considered an influential point.

- If there are influential points, briefly describe strategies to deal with them in your regression analysis.

Since there is only one influential point, one good strategy is to simply delete the Alaska observation to make our regression model more accurate.

12. Lastly, let's examine the Variance Inflation Factor (VIF) used to determine if the predictor variables in the model are correlated with each other.

Let's start by manually calculating VIF for the variable Expend.

- Begin by fitting a model with Expend as the response variable and the other predictor variables in model_select_aic as the predictors.

```
model <- lm(Expend ~ Years + Public + Rank, data=sat_scores)
summary(model)
```

```
Output:
Call:
lm(formula = Expend ~ Years + Public + Rank, data = sat_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0866 -3.9495 -0.1809  2.3098 25.1092

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.23862   25.54114  -0.401  0.69037
Years         2.19154    1.27212   1.723  0.09165 .
Public        0.25256    0.09047   2.792  0.00761 **
Rank         -0.28539    0.12423  -2.297  0.02620 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.636 on 46 degrees of freedom
Multiple R-squared:  0.2102,    Adjusted R-squared:  0.1587
F-statistic: 4.081 on 3 and 46 DF,  p-value: 0.01189
```

- Calculate R2 for this model.

```
summary(model)$r.squared
```

Output: 0.2102009

- Use this R2 to calculate VIF for Expend.

VIF = 1 / (1 - R^2) = 1 / (1 - (0.2102009)^2) = 1.04622

- Does Expend appear to be highly correlated with any other predictor variables? Briefly explain.

Expend does not appear to be highly correlated with any other predictor variables because the VIF value is very close to 1.

13. Now, let's use the vif function in the rms package to calculate VIF for all of the variables in the model. You can use the tidy function to output the results neatly in a data frame. Are there any obvious concerns with multicollinearity in this model? Briefly explain.

```
vif(model_select_aic)
```

Using the vif function, I got these results for the VIF of each variable:

```
   Years   Public   Expend     Rank
1.301929 1.426831 1.266145 1.129034
```

There are no obvious concerns with multicollinearity, since all of the variables have a VIF relatively close to 1 (all are less than 1.5). However, the variable with the highest VIF is Public.