lab-06: Model Selection + Diagnostics

Justin Chan

2/21/2022

Packages

Loading required packages for the lab:

```
library(tidyverse)
library(knitr)
library(broom)
library(leaps)
library(rms)
library(Sleuth3)
```

Exercises

Part I: 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)
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -94.7      212.       -0.448  0.657
## 2 Takers       -0.480      0.694    -0.692  0.493
## 3 Income       -0.00820    0.152    -0.0538 0.957
## 4 Years        22.6        6.31      3.58   0.000866
## 5 Public       -0.464      0.579    -0.802  0.427
## 6 Expend        2.21       0.846     2.61   0.0123
## 7 Rank          8.48       2.11      4.02   0.000230
```

1. We will use the regsubsets function in the leaps R package to perform backward selection on multiple linear regression models with Adj.R2 or BIC as the selection criteria.

```r
model_select <- regsubsets(SAT ~ Takers + Income + Years + Public + Expend + Rank , data = sat_scores, n
select_summary <- summary(model_select)
coef(model_select, 1:6)
```

```
## [[1]]
## (Intercept)        Rank
##  183.418763    9.557949
##
## [[2]]
## (Intercept)       Years        Rank
## -243.930900   27.382901    9.351603
##
## [[3]]
## (Intercept)       Years      Expend        Rank
## -303.724295   26.095227    1.860866    9.825794
##
## [[4]]
## (Intercept)       Years      Public      Expend        Rank
## -204.598232   21.890482   -0.663798    2.241640   10.003169
##
## [[5]]
##  (Intercept)       Takers        Years       Public       Expend         Rank
## -100.4736967   -0.4620796   22.6688085   -0.4522606    2.1859091    8.4964099
##
## [[6]]
##   (Intercept)         Takers         Income          Years         Public
## -94.659108883   -0.480080120   -0.008195013   22.610081908   -0.464152292
##        Expend           Rank
##   2.212004850    8.476216985
```

```r
select_summary$adjr2
```

```
## [1] 0.7695367 0.8405479 0.8627047 0.8661268 0.8649009 0.8617684
```

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

```r
select_summary$bic
```

```
## [1] -66.59010 -82.14815 -86.79191 -85.24089 -81.99674 -78.08808
```

3. 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.

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

```
## Start:  AIC=333.58
## SAT ~ Takers + Income + Years + Public + Expend + Rank
```

```
##
##          Df Sum of Sq   RSS    AIC
## - Income  1       2.0 29844 331.59
## - Takers  1     332.4 30175 332.14
## - Public  1     445.8 30288 332.32
## <none>                29842 333.58
## - Expend  1    4744.9 34587 338.96
## - Years   1    8897.8 38740 344.63
## - Rank    1   11223.0 41065 347.54
##
## Step:  AIC=331.59
## SAT ~ Takers + Years + Public + Expend + Rank
##
##          Df Sum of Sq   RSS    AIC
## - Takers  1     401.3 30246 330.25
## - Public  1     495.5 30340 330.41
## <none>                29844 331.59
## - Expend  1    6904.4 36749 339.99
## - Years   1    9219.7 39064 343.05
## - Rank    1   11645.9 41490 346.06
##
## Step:  AIC=330.25
## SAT ~ Years + Public + Expend + Rank
##
##          Df Sum of Sq    RSS    AIC
## <none>                 30246 330.25
## - Public  1      1462  31708 330.62
## - Expend  1      7343  37589 339.12
## - Years   1      8837  39083 341.07
## - Rank    1    184786 215032 426.33
```

```
tidy(model_select_aic)
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 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
```

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.

The models do not have the same predictors where the BIC has three predictors and the AIC and Adj.R2 have four predictors based on the models created above. The selection criterion that resulted in the fewest number of parameters is BIC with three predictors. No, this is not what I would expect completely. I would expect AIC and BIC would have fewer predictors since they help determine the best possible model without overpredicting or adding more variables that may add more noise than help by factoring the log likelihood. However, BIC only had the smallest amount predictors based on the smallest BIC value.

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 prediictiosn 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.

```
df_model_prediction_statistics <- augment(model_select_aic)
df_model_prediction_statistics <- df_model_prediction_statistics %>%
  mutate(obs_num = row_number())
head(df_model_prediction_statistics, 5)
```

```
## # A tibble: 5 x 12
##     SAT Years Public Expend  Rank .fitted .resid   .hat .sigma .cooksd
##   <int> <dbl>  <dbl>  <dbl> <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
## 1  1088  16.8   87.8   25.6  89.7   1059.  28.7  0.100    25.8 0.0304
## 2  1075  16.1   86.2   20.0  90.6   1041.  34.0  0.0788   25.7 0.0320
## 3  1068  16.6   88.3   20.6  89.8   1044.  24.0  0.0894   25.9 0.0185
## 4  1045  16.3   83.9   27.1  86.3   1021.  24.4  0.0585   25.9 0.0117
## 5  1045  17.2   83.6   21.0  88.5   1050.  -4.99 0.113    26.2 0.00106
## # ... with 2 more variables: .std.resid <dbl>, obs_num <int>
```

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.

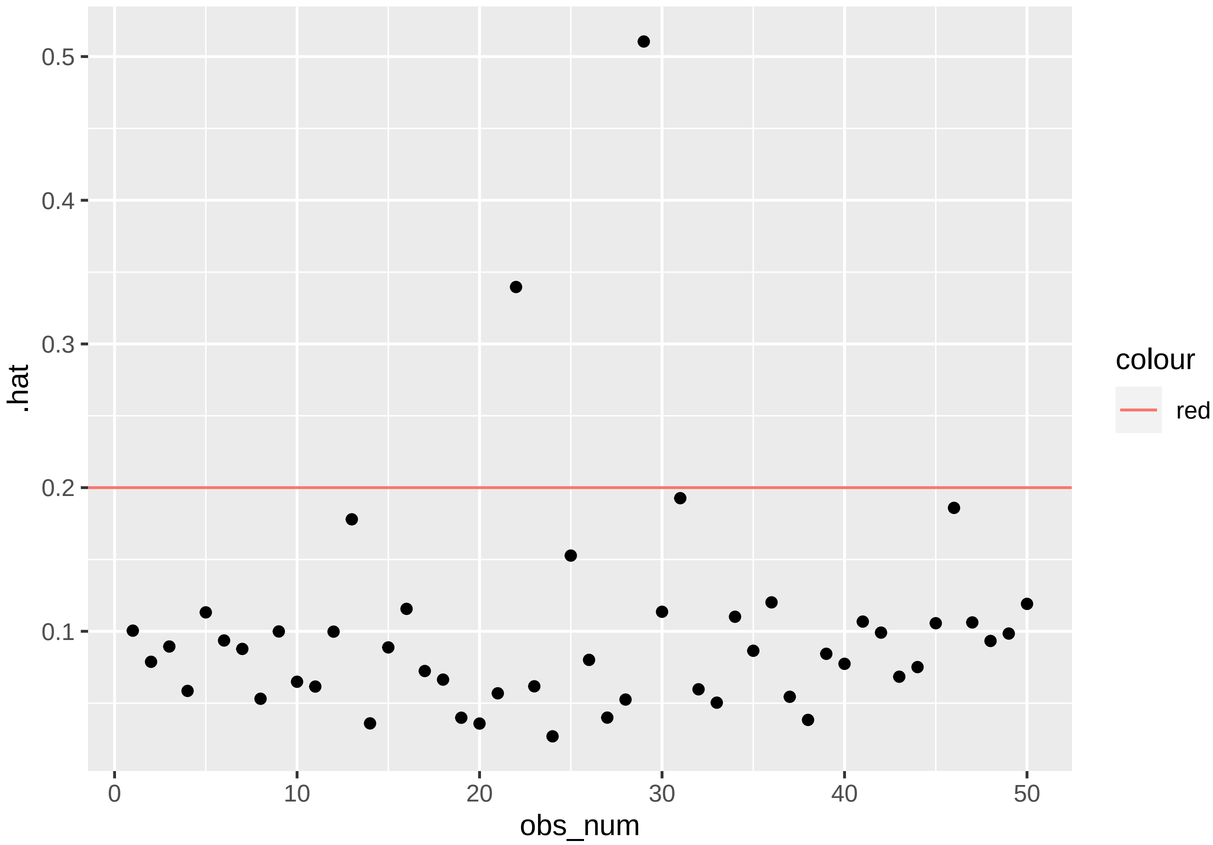
The threshold we should use to determine if the observations in this dataset have high leverage if its over 2 times the average leverage for all observations from the lecture. The equation to calculate the threshold:

$$h_i > \frac{2(p+1)}{n} > \frac{2(4+1)}{50} > 0.2$$

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.

```
leverage_threshold <- 2*(4+1)/nrow(df_model_prediction_statistics)

ggplot(data = df_model_prediction_statistics, aes(x = obs_num, y = .hat)) +
  geom_point() +
  geom_hline(aes(yintercept = leverage_threshold, colour = "red"))
```



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.

```
df_model_prediction_statistics %>% filter(.hat > leverage_threshold) %>%
  select(obs_num, Years, Public, Expend, Rank)
```

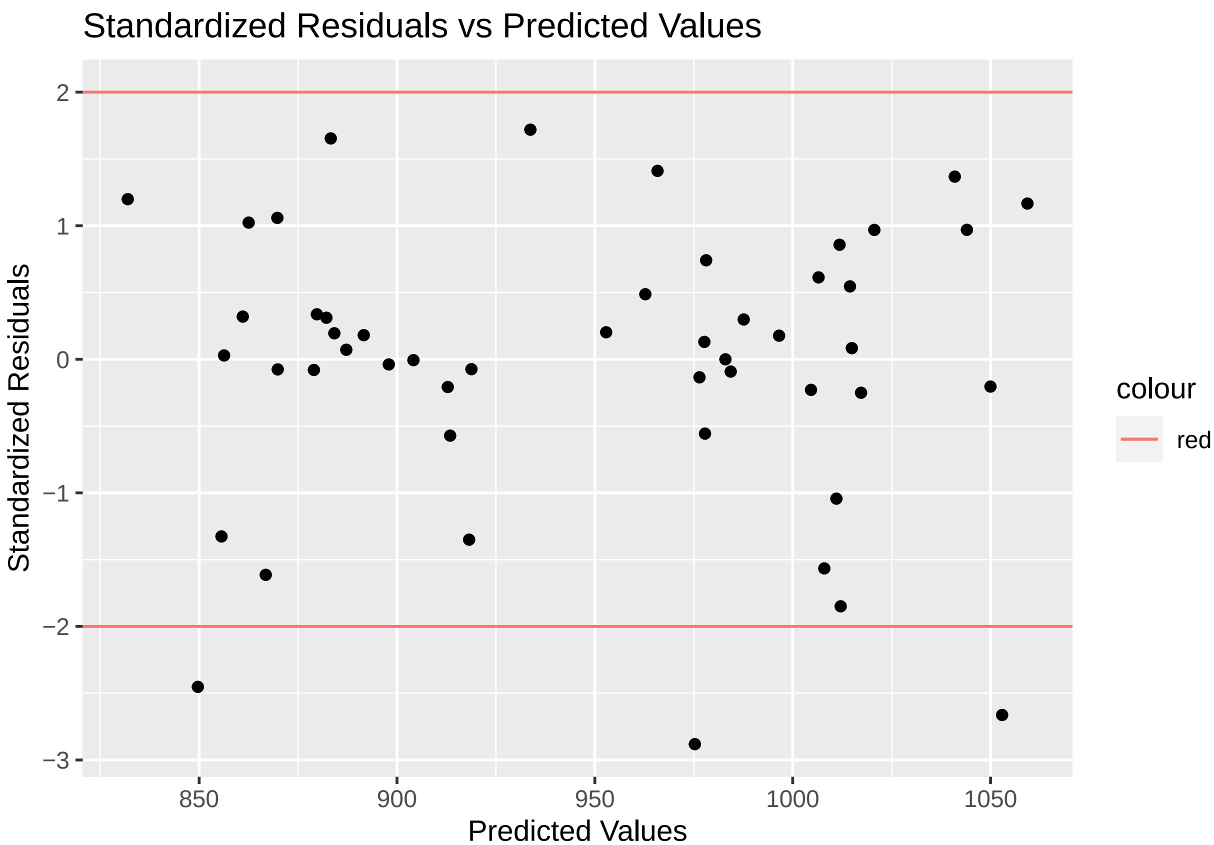
```
## # A tibble: 2 x 5
##   obs_num Years Public Expend  Rank
##     <int> <dbl>  <dbl>  <dbl> <dbl>
## 1      22  16.8   44.8   19.7  82.9
## 2      29  15.3   96.5   50.1  79.6
```

The states in the dataset that are considered high leverage is Louisiana and Alaska which are the rows 22 and 29 in the sat_scores dataset.

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=-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.

```
ggplot(data = df_model_prediction_statistics, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(aes(yintercept = 2, colour = "red")) +
  geom_hline(aes(yintercept = -2, colour = "red")) +
  labs(title = "Standardized Residuals vs Predicted Values",
```

```
x = "Predicted Values",
y = "Standardized Residuals")
```



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.

```
df_model_prediction_statistics %>% filter(2 < abs(.std.resid)) %>%
  select(obs_num, Years, Public, Expend, Rank)
```

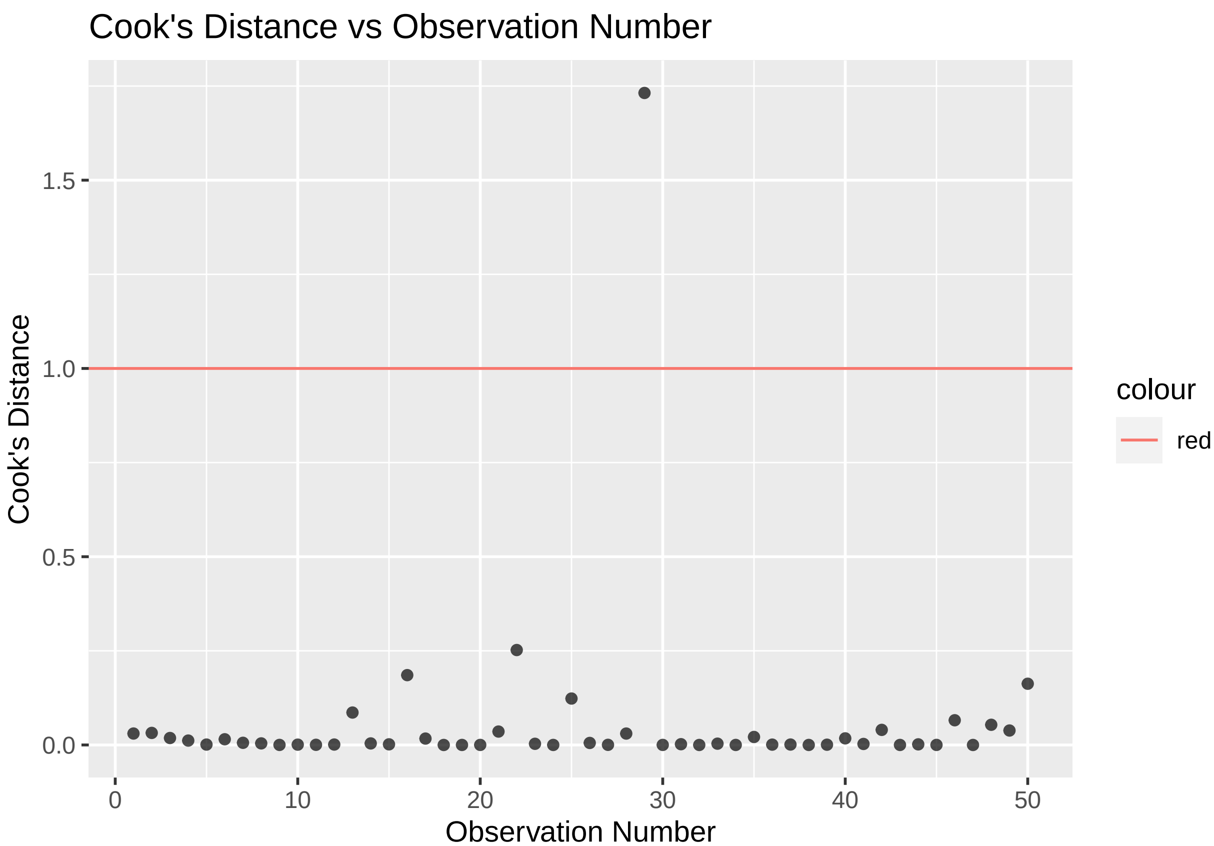
```
## # A tibble: 3 x 5
##   obs_num Years Public Expend  Rank
##     <int> <dbl>  <dbl>  <dbl> <dbl>
## 1      16  16.8   67.9   15.4  90.1
## 2      29  15.3   96.5   50.1  79.6
## 3      50  15.4   88.1   15.6  74
```

The states in the dataset that are considered to have standardized residuals with large magnitude are Mississippi, Alaska, and North Carolina which are the rows 16, 29 and 50 in the sat_scores dataset.

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.

- Which states (if any) are considered to be influential points?
- If there are influential points, briefly describe strategies to deal with them in your regression analysis.

```
ggplot(data = df_model_prediction_statistics, aes(x = obs_num, y = .cooksd)) +
  geom_point(alpha = 0.7) +
  geom_hline(aes(yintercept = 1, colour = "red")) +
  labs(title = "Cook's Distance vs Observation Number",
       x = "Observation Number",
       y = "Cook's Distance")
```




```
df_model_prediction_statistics %>% filter(1 < .cooksd) %>%
  select(obs_num, Years, Public, Expend, Rank)
```

```
## # A tibble: 1 x 5
##   obs_num Years Public Expend  Rank
##     <int> <dbl>  <dbl>  <dbl> <dbl>
## 1      29  15.3   96.5   50.1  79.6
```

The state in the dataset that are considered to be influential point is Alaska which is row 29 in the sat_scores dataset. One strategy to deal with these influencial points or ways to drop the point based on predictor variables if it is meaningful to drop the observation given the context of the problem, build a model with a smaller range of predictor variables and mention this in the write up. Other strategies to deal with these influential points is transformations or increasing the sample size by collecting more data.

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. - Calculate R2 for this model. - Use this R2 to calculate VIF for Expend. - Does Expend appear to be highly correlated with any other predictor variables? Briefly explain.

```r
expend_model <- lm(Expend ~ Years + Public + Rank, data = sat_scores)
summary(expend_model)$r.squared
```

```
## [1] 0.2102009
```

$$VIF = \frac{1}{1 - R^2_{expend}} = \frac{1}{1 - 0.2102} = 1.27$$

The Expend does not appearly to be highly correlated with any other predictor variables since VIF value, 1.26 is not large enough to show mutlicollinearity with other variables. Often, VIF values over 10 indicate concerning mutlicollinearity where there is high correlation between two or more explanatory variables which often occurs in smaller sample sizes. In this model, mutlicollinearity does not seem to be an issue with the variable extend that has a VIF much smaller than 10 as previously mentioned.

12. 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.

```r
tidy(vif(model_select_aic))
```

```
## Warning: 'tidy.numeric' is deprecated.
## See help("Deprecated")

## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.

## # A tibble: 4 x 2
##   names      x
##   <chr>  <dbl>
## 1 Years   1.30
## 2 Public  1.43
## 3 Expend  1.27
## 4 Rank    1.13
```

No, there does not seem an issue of multicollinearity in this model, model_select_aic after using the function vif() from the rms package which has all the VIF explanatory values in the model to be less than 1.43. If any of these VIF values were over 10, then we would most likely have a problem with mutlicollinearity but, there does not seem to be these obvious concerns at this first glance/analysis of the model.