The goal of today's lab is to gain practice with model selection and model diagnostic procedures in R. This includes

Packages

You will need the following packages for today's lab:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
## Loading required package: Hmisc
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
##
## Loading required package: SparseM
##
## Attaching package: 'SparseM'
##
## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(Sleuth3) #case1201 data
```

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 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
head(sat_scores)
```

```
##         State  SAT Takers Income Years Public Expend Rank
## 1        Iowa 1088      3    326 16.79   87.8  25.60 89.7
## 2 SouthDakota 1075      2    264 16.07   86.2  19.95 90.6
## 3 NorthDakota 1068      3    317 16.57   88.3  20.62 89.8
## 4      Kansas 1045      5    338 16.30   83.9  27.14 86.3
## 5    Nebraska 1045      5    293 17.25   83.6  21.05 88.5
## 6     Montana 1033      8    263 15.91   93.7  29.48 86.4
```

```
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
```

```
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)
coef(model_select, which.max(select_summary$adjr2)) #display coefficients
```

```
## (Intercept)       Years      Public      Expend        Rank
## -204.598232   21.890482   -0.663798    2.241640   10.003169
```

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

```
coef(model_select, which.min(select_summary$bic)) #display coefficients
```

```
## (Intercept)       Years      Expend        Rank
## -303.724295   26.095227    1.860866    9.825794
```

```
Type `??step` in the console for more information about the `step` function. The output from the `step`
```

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.

```
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
```

4. Compare the final models selected by $Adj.R^2$, 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.

    ANSWER: No, only $Adj.R^2$ and AIC have same number of predictors. BIC have the fewest number of predictors because it penalizes the number of parameters more strongly than AIC does.

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 predictions 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.

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

```
## # A tibble: 5 x 12
##     SAT Years Public Expend  Rank .fitted .resid   .hat .sigma .cooksd .std.re~1
##   <int> <dbl>  <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      1.17
## 2  1075  16.1   86.2   20.0  90.6   1041.  34.0  0.0788   25.7 0.0320      1.37
## 3  1068  16.6   88.3   20.6  89.8   1044.  24.0  0.0894   25.9 0.0185      0.969
## 4  1045  16.3   83.9   27.1  86.3   1021.  24.4  0.0585   25.9 0.0117      0.969
## 5  1045  17.2   83.6   21.0  88.5   1050.  -4.99 0.113    26.2 0.00106    -0.204
## # ... with 1 more variable: obs_num <int>, and abbreviated variable name
## #   1: .std.resid
```

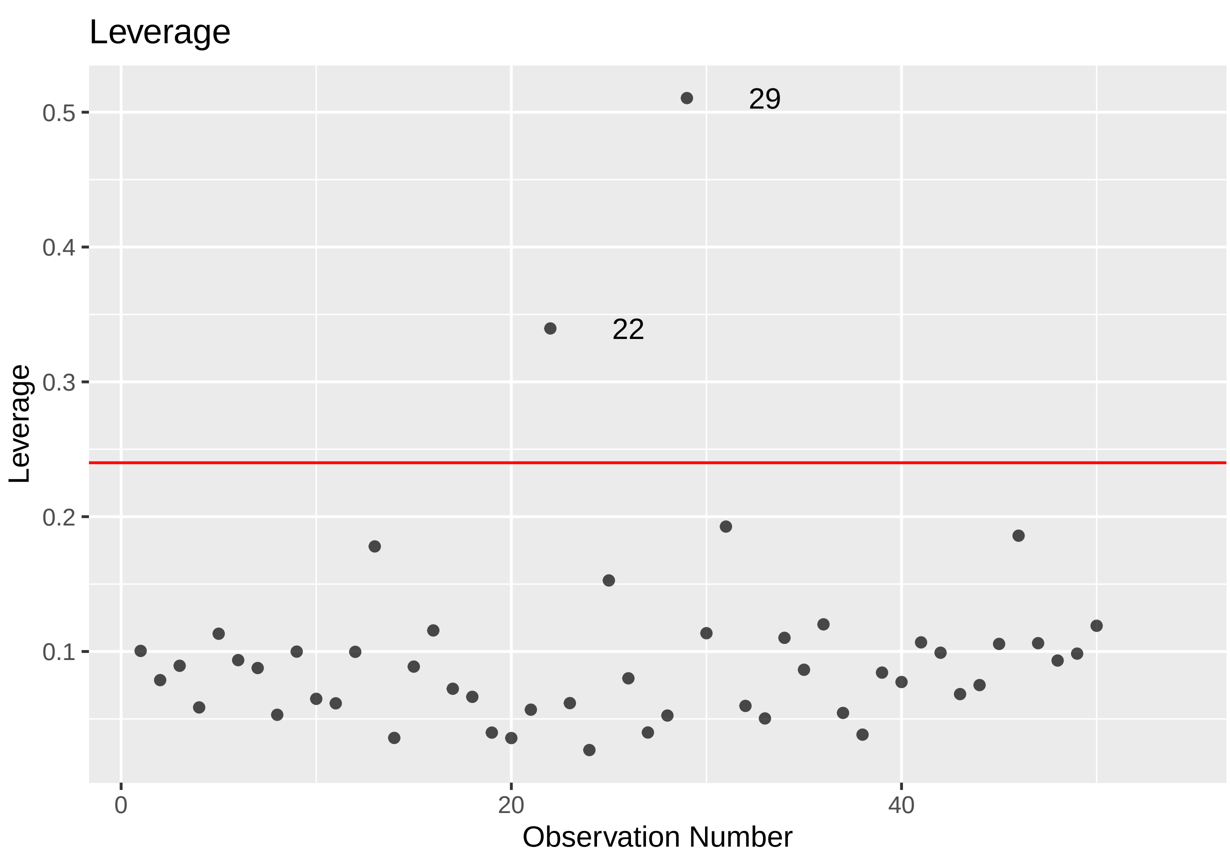
6. Let's examine the leverage for each observation. Based on the lecture notes, what threshold should we use to determine if observations in this dataset have high leverage? *Report the value and show the quation you used to calculate it.*

```
(leverage_threshold <- 2*(5+1)/nrow(model_stats))
```

```
## [1] 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.*

```
ggplot(data = model_stats, aes(x = obs_num, y = .hat)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = leverage_threshold, color = "red")+
  labs(x = "Observation Number",y = "Leverage",title = "Leverage") +
  geom_text(aes(label=ifelse(.hat > leverage_threshold, as.character(obs_num), "")), nudge_x = 4)
```



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`.*

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

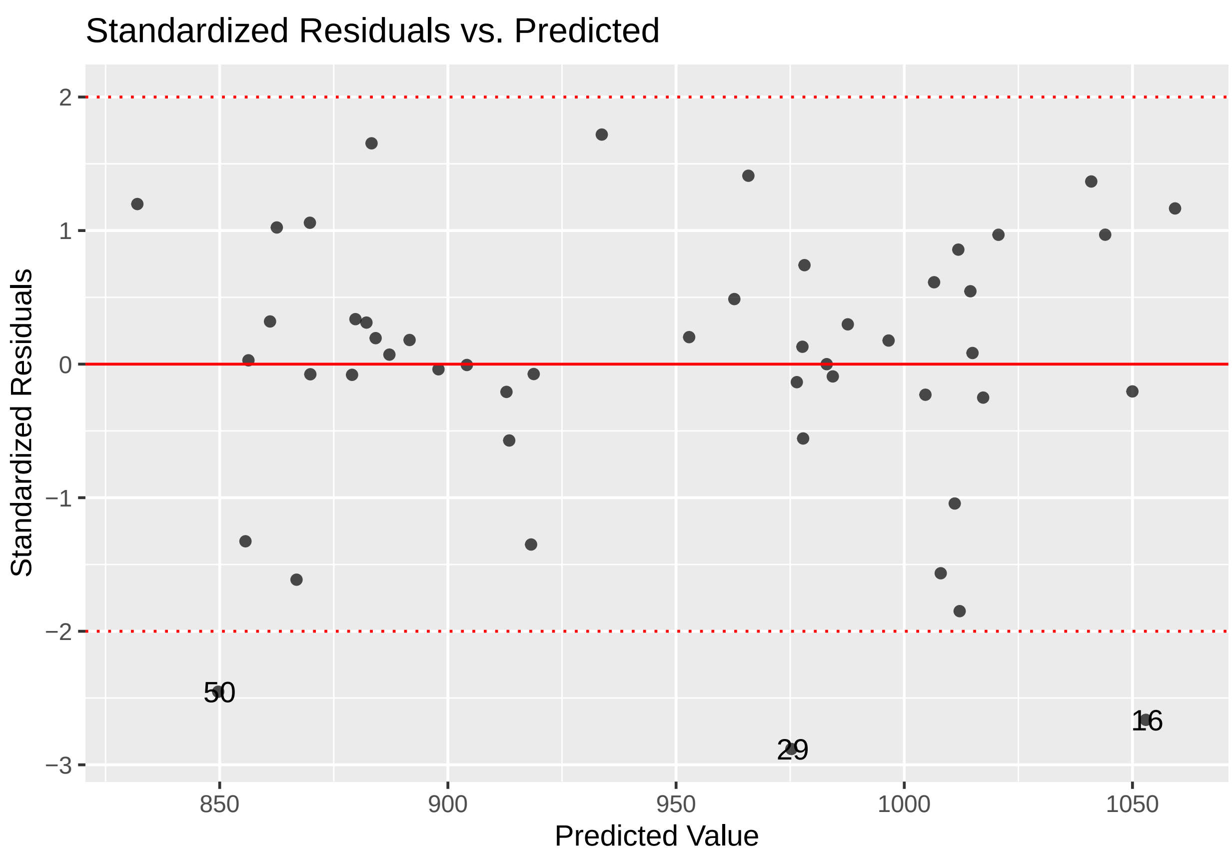
```
## # A tibble: 2 x 5
##   obs_num Years Public Expend  Rank
##     <int> <dbl>  <dbl>  <dbl> <dbl>
## 1      22  16.8   44.8   19.7  82.9
## 2      29  15.3   96.5   50.1  79.6
```

```r
sat_scores[c(22,29), 'State']
```

```
## [1] Louisiana Alaska
## 50 Levels: Alabama Alaska Arizona Arkansas California Colorado ... Wyoming
```

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.*

```r
ggplot(data = model_stats, aes(x = .fitted,y = .std.resid)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0,color = "red") +
  geom_hline(yintercept = -2,color = "red",linetype = "dotted") +
  geom_hline(yintercept = 2,color = "red",linetype = "dotted") +
  labs(x ="Predicted Value",y ="Standardized Residuals",title = "Standardized Residuals vs. Predicted")
  geom_text(aes(label = ifelse(abs(.std.resid) > 2,as.character(obs_num),"")), nudge_x = 0.3)
```



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`.*

```r
model_stats %>% filter(.std.resid > 2 | .std.resid < -2) %>%
  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
```

```r
sat_scores[c(16, 29, 50), 'State']
```

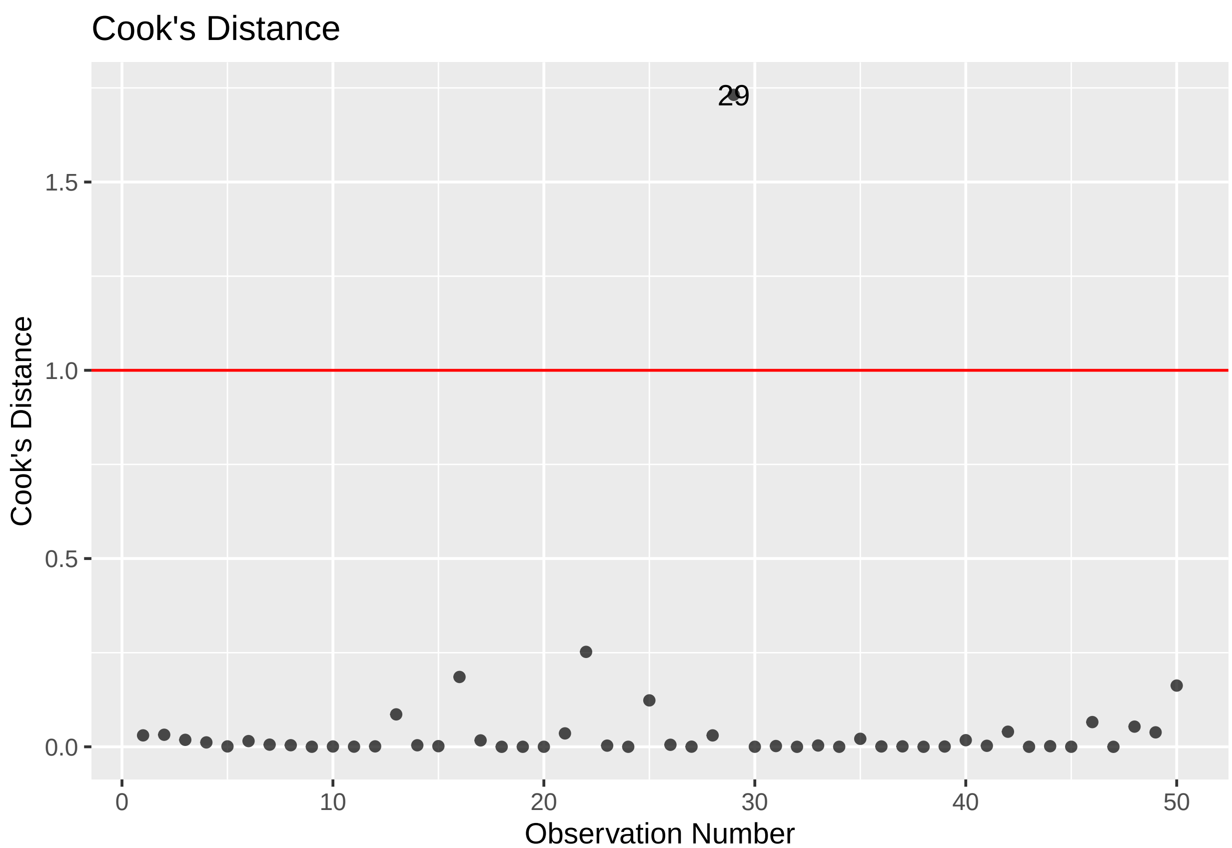
```
## [1] Mississippi   Alaska        SouthCarolina
## 50 Levels: Alabama Alaska Arizona Arkansas California Colorado ... Wyoming
```

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.

    ANSWER: Alaska is considered to be influential point. Since we want to analyze the risk factors of SAT scores in United States, I'll keep this influential point because Alaska is also a state in the US. If the observation has some errors and we intend to build a model on a smaller range of the predictor variables, I'll drop it.

```r
ggplot(data = model_stats, aes(x = obs_num, y = .cooksd)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept=1,color = "red")+
  labs(x= "Observation Number",y = "Cook's Distance",title = "Cook's Distance") +
  geom_text(aes(label = ifelse(.cooksd > 1,as.character(obs_num),"")))
```




```r
sat_scores[29, 'State']
```

```
## [1] Alaska
## 50 Levels: Alabama Alaska Arizona Arkansas California Colorado ... Wyoming
```

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.

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{x_j|x_{-j}}}$$

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 R^2 for this model.
- Use this R^2 to calculate VIF for `Expend`.
- Does `Expend` appear to be highly correlated with any other predictor variables? Briefly explain.

No, since VIF is 1.266 which is relatively low.

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

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

```r
1 / (1 - r_squared)
```

```
## [1] 1.266145
```

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.

ANSWER: There's no obvious concerns with multicollinearity since all variables have a relatively low VIF.

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

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

```
## # 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
```

Submitting the Assignment

As before, what the instructor is going to check is your repo. Make sure to produce a pdf, and include it in your repo with the name Lab06.pdf. Also, include a folder called raw_data where the original data should be stored, and another folder called mod_data where the final version of your data table should be stored. Finally, a Readme.md should be created with a short description of this lab and the data.

Repo: https://github.com/ysun155/STAT108-labs/tree/main/Lab6