lab6

GITHUB: https://github.com/theeho/lab6

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
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
```

Exercise 1/2: Note: Was not sure how to use coef function so I used tidy function. I think it works.

```
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
```

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


coef(model_select, 6) #display coefficients
```

```
##   (Intercept)        Takers        Income         Years        Public
## -94.659108883  -0.480080120  -0.008195013  22.610081908  -0.464152292
##        Expend          Rank
##   2.212004850   8.476216985
```

```
BIC_coef <- tidy(model_select) %>% pull(BIC)
adjr_coef <- tidy(model_select) %>% pull(adj.r.squared)
BIC_coef
```

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

```
adjr_coef
```

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

Exercise 3:

```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
```

```r
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
```

Exercise 4: The models do not have the same number of predictors. The AIC has the least number of predictors at 4 while BIC and adjr2 have 6 predictors. This is expected because AIC has a greater penalty for more predictors compared to BIC and adjr2. Exercise 5:

```
sat_aug <- augment(model_select_aic) %>%
      mutate(obs_num = row_number())
head(sat_aug, n=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>
```

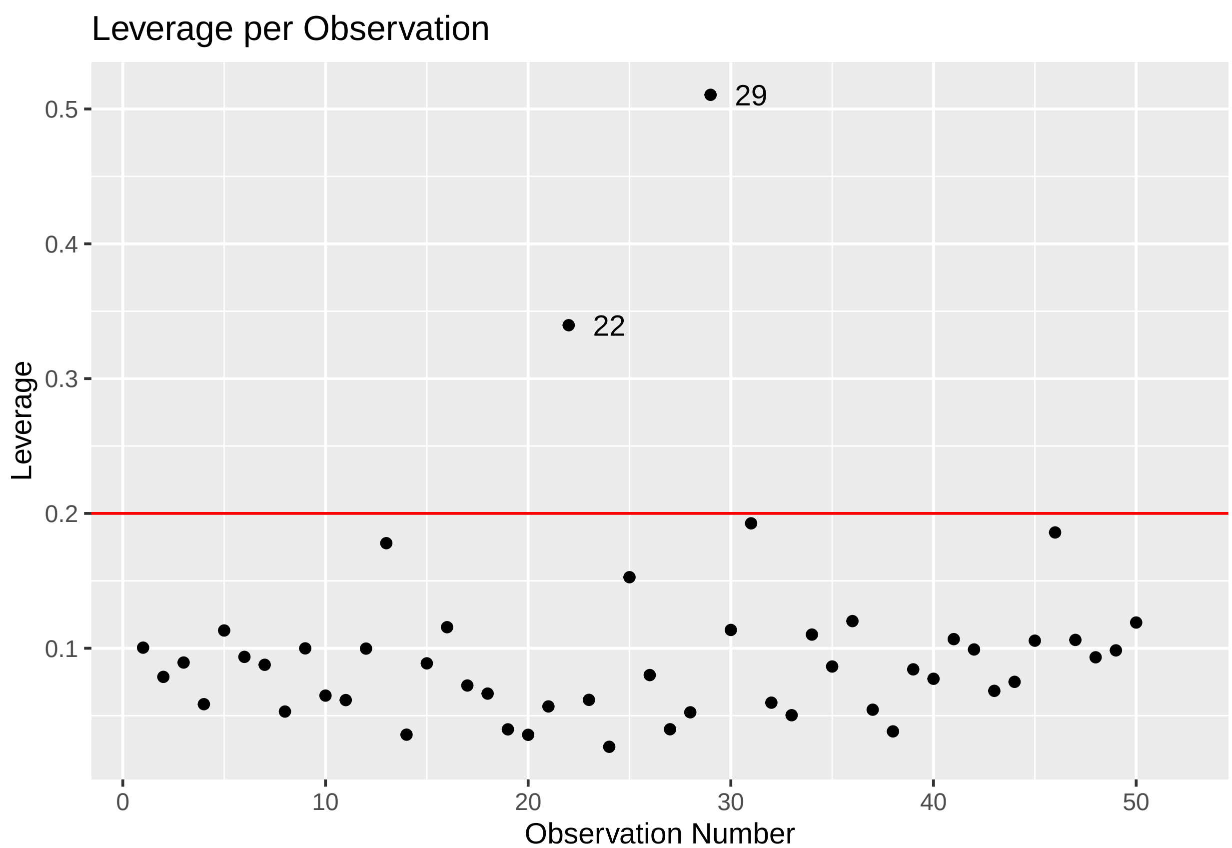
Exercise 6: Based on lecture notes, we should use 2(p+1)/n as our leverage threshold.

Exercise 7:

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

```
## [1] 0.2
```

```
ggplot(data = sat_aug, aes(x = obs_num, y = .hat)) +
  geom_point() +
  geom_hline(yintercept = leverage_threshold, color = "red")+
  labs(x = "Observation Number",y = "Leverage",title = "Leverage per Observation") +
  geom_text(aes(label=ifelse(.hat > leverage_threshold, as.character(obs_num), "")), nudge_x = 2)
```



Exercise 8:

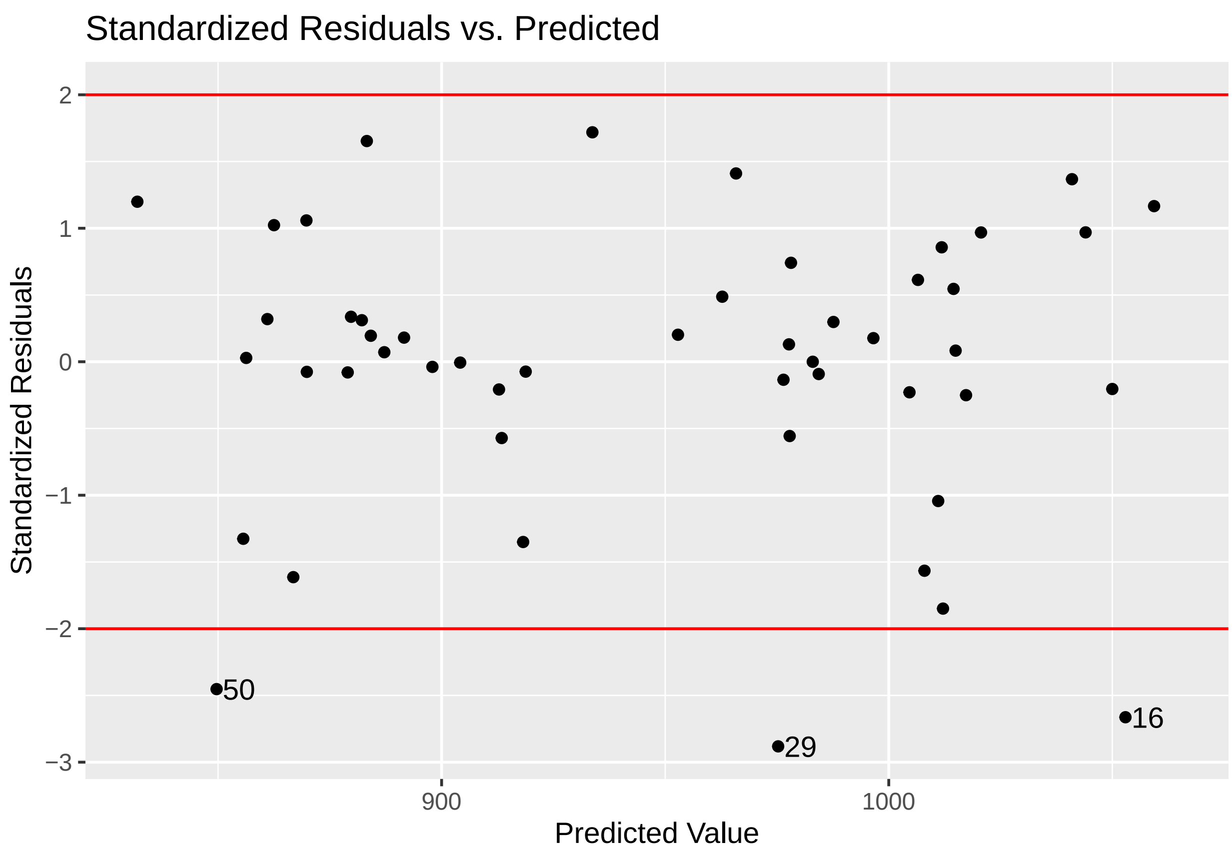
```
obnum_s <- sat_scores %>% mutate(obs_num = row_number())
high_lev <- filter(obnum_s, obs_num == 22 | obs_num == 29)
head(high_lev, n = 2)
```

```
##       State SAT Takers Income Years Public Expend Rank obs_num
## 1 Louisiana 975      5    394 16.85   44.8  19.72 82.9      22
## 2    Alaska 923     31    401 15.32   96.5  50.10 79.6      29
```

It appears that Alaska and Louisiana are the two high leverage points.

Exercise 9:

```
ggplot(data = sat_aug, aes(x = .fitted,y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = -2,color = "red") +
  geom_hline(yintercept = 2,color = "red") +
  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 = 5)
```



Exercise 10:

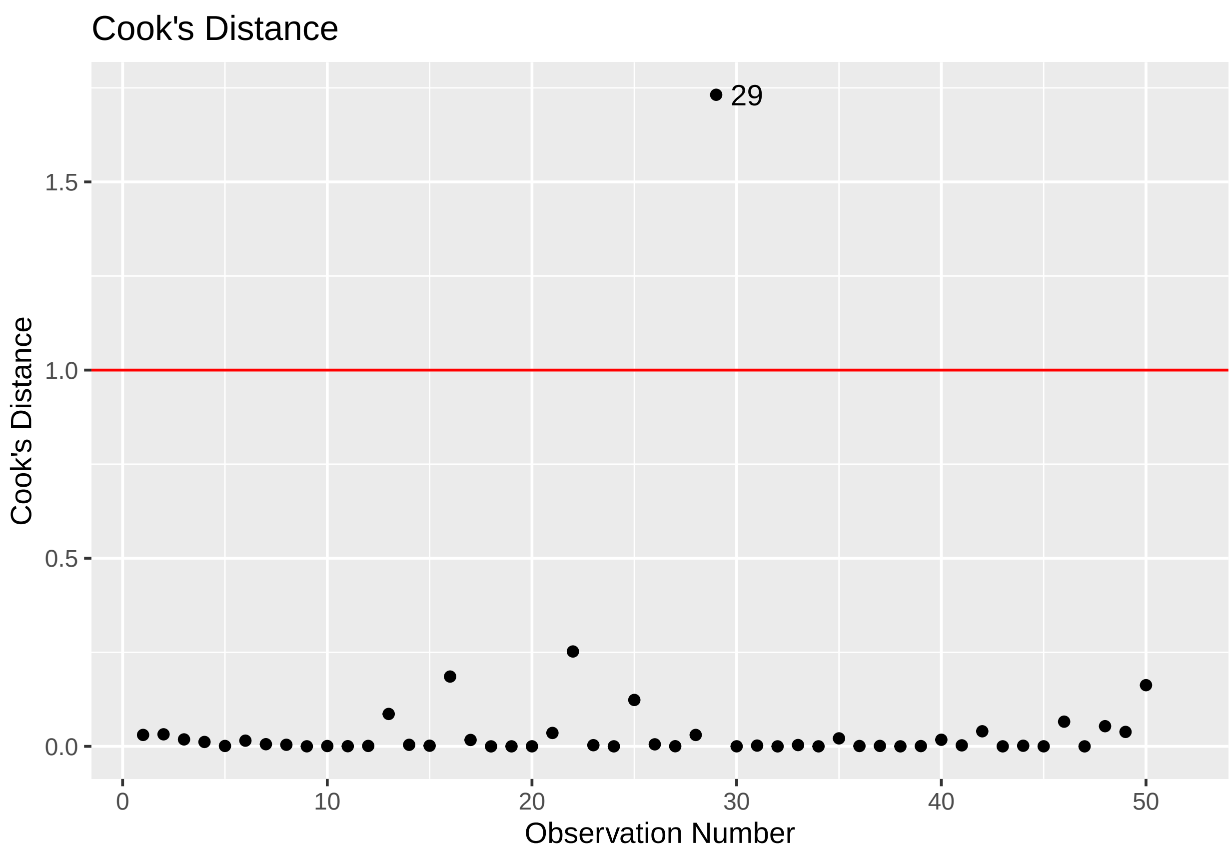
```
high_res <- filter(obnum_s, obs_num == 16 | obs_num == 29 | obs_num == 50)
head(high_res, n = 3)
```

```
##            State SAT Takers Income Years Public Expend Rank obs_num
## 1    Mississippi 988      3    315 16.76   67.9  15.36 90.1      16
## 2         Alaska 923     31    401 15.32   96.5  50.10 79.6      29
## 3 SouthCarolina 790     48    214 15.42   88.1  15.60 74.0      50
```

It appears that Mississippi, Alaska, and South Carolina have large standardized residuals.

Exercise 11:

```
ggplot(data = sat_aug, aes(x = obs_num, y = .cooksd)) +
  geom_point() +
  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),"")), nudge_x =1.5)
```



It appears Alaska has a high cooks distance and is therefore considered an influential point. I think the best practice would be to check the model with and without this point, and also determine if the outlier is due to the predictor variables or response variables.

Exercise 12

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

```
##
## Call:
## lm(formula = Expend ~ Rank + Years + Public, data = sat_aug)
##
## 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
## Rank         -0.28539    0.12423  -2.297  0.02620 *
## Years         2.19154    1.27212   1.723  0.09165 .
## Public        0.25256    0.09047   2.792  0.00761 **
## ---
## 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
```

```r
tidy(vif(sat_modelr))
```

```
## 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: 3 x 2
##   names      x
##   <chr>  <dbl>
## 1 Rank    1.01
## 2 Years   1.22
## 3 Public  1.22
```

Because the VIF valuesfor each paramater are small, Expend does not appear to be correlated.

```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
```

In this model the VIC values for all the paramaters are also small. None of the parameters appear to be correlated.