Hw5

Stone Cai

2024-03-21

```r
library (faraway)
```

```
## Warning: package 'faraway' was built under R version 4.3.2
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.2

##
## Attaching package: 'car'

## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```r
data_hw <- read.csv("/Users/stone/Documents/Stat 408/Homework Data Files/Hw5/dataHW5.csv")

summary(data_hw)
```

```
##        y                   x1                x2               x3
##  Min.   :      0.1   Min.   :-1.0000   Min.   :0.010   Min.   :-1.8200
##  1st Qu.:      6.7   1st Qu.:-0.4925   1st Qu.:0.620   1st Qu.:-0.2425
##  Median :    364.3   Median : 0.0150   Median :1.235   Median : 0.2400
##  Mean   : 122118.2   Mean   : 0.0186   Mean   :1.146   Mean   : 0.4038
##  3rd Qu.:  19919.4   3rd Qu.: 0.5125   3rd Qu.:1.623   3rd Qu.: 1.2125
##  Max.   :1295689.1   Max.   : 0.9800   Max.   :1.990   Max.   : 3.3400
##        x4                x5                x6                x7
##  Min.   :-2.7600   Min.   :-4.9400   Min.   :-214.72   Min.   :-9.580
##  1st Qu.: 0.5225   1st Qu.:-2.2200   1st Qu.: -39.53   1st Qu.:-7.635
##  Median : 1.3050   Median : 0.6600   Median :  28.47   Median :-5.680
##  Mean   : 1.1187   Mean   : 0.3826   Mean   :  15.82   Mean   :-5.196
##  3rd Qu.: 1.8825   3rd Qu.: 3.2325   3rd Qu.:  68.45   3rd Qu.:-2.723
##  Max.   : 3.8500   Max.   : 4.9000   Max.   : 230.98   Max.   : 0.800
##        x8               x9              x10
##  Min.   : 0.460   Min.   :1.020   Min.   :-30.830
```

```
##  1st Qu.: 2.250   1st Qu.:2.550   1st Qu.:-10.145
##  Median : 3.855   Median :5.415   Median : -3.090
##  Mean   : 4.578   Mean   :5.442   Mean   : -3.309
##  3rd Qu.: 6.325   3rd Qu.:8.300   3rd Qu.:  3.257
##  Max.   :12.240   Max.   :9.910   Max.   : 22.950
```

```r
##plot (data_hw)

full_mod<- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10,data= data_hw)
summary(full_mod)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10, data = data_hw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -331585 -169208  -30433   83876  849386
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -212075.7   104193.3  -2.035   0.0448 *
## x1            16058.0    49151.6   0.327   0.7447
## x2           366284.2   501239.7   0.731   0.4668
## x3            19026.4    26056.4   0.730   0.4672
## x4            28001.8    29169.8   0.960   0.3397
## x5            -1604.7     8949.7  -0.179   0.8581
## x6              -41.5      295.9  -0.140   0.8888
## x7            87710.3   100322.4   0.874   0.3843
## x8             1644.0     8945.5   0.184   0.8546
## x9            61674.6     9820.4   6.280 1.21e-08 ***
## x10            3351.4     2714.3   1.235   0.2202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 256000 on 89 degrees of freedom
## Multiple R-squared:  0.3476, Adjusted R-squared:  0.2743
## F-statistic: 4.742 on 10 and 89 DF,  p-value: 1.959e-05
```

```r
################## create scatterplot matrix
```

First I attached raw hw5 data set and conducted an initial numerical summary and a full model.Almost immediately I notice several initial problems. It seems like the Y response variable is heavily right skewed with a vast majority of observations having very small values and few values having extremely large values. Also, almost every single p-value of a predictor is not significant which suggests there may be some structural problem.

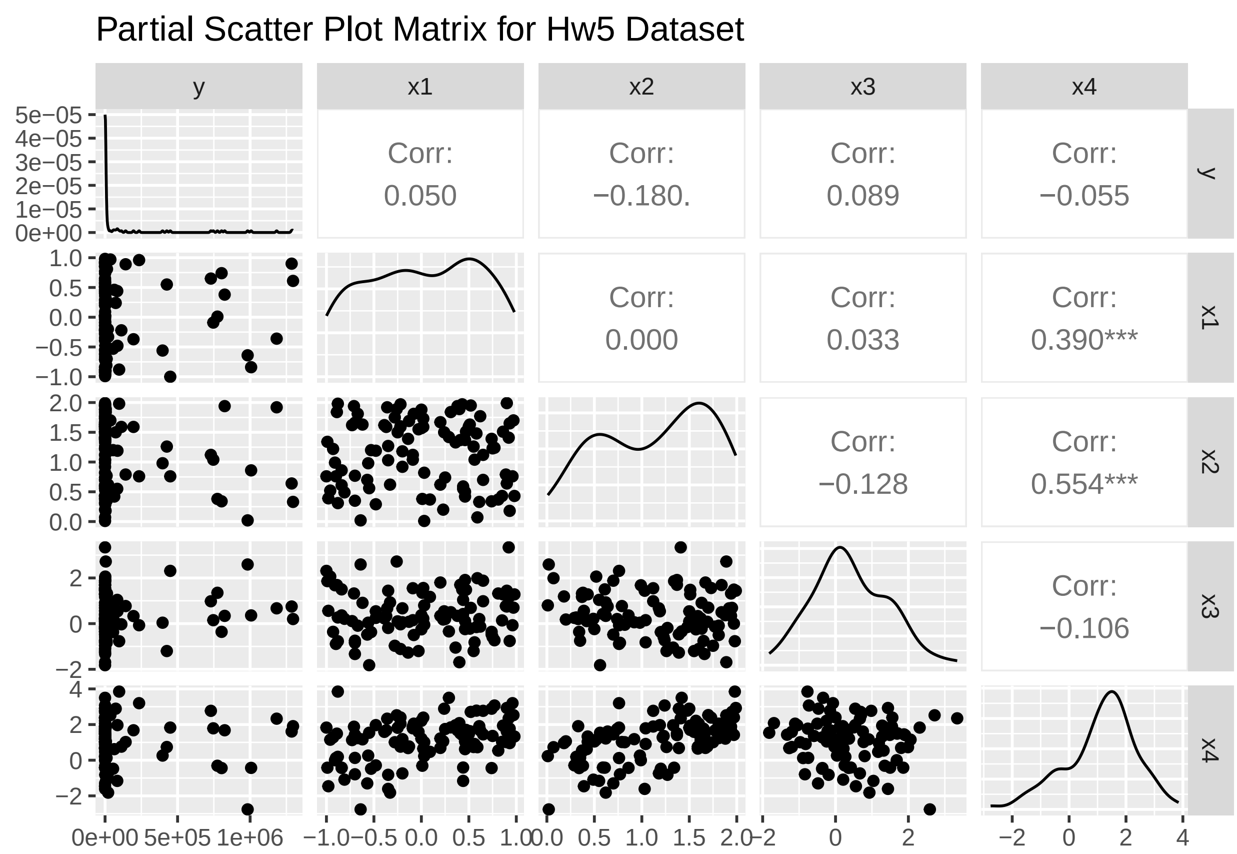
```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

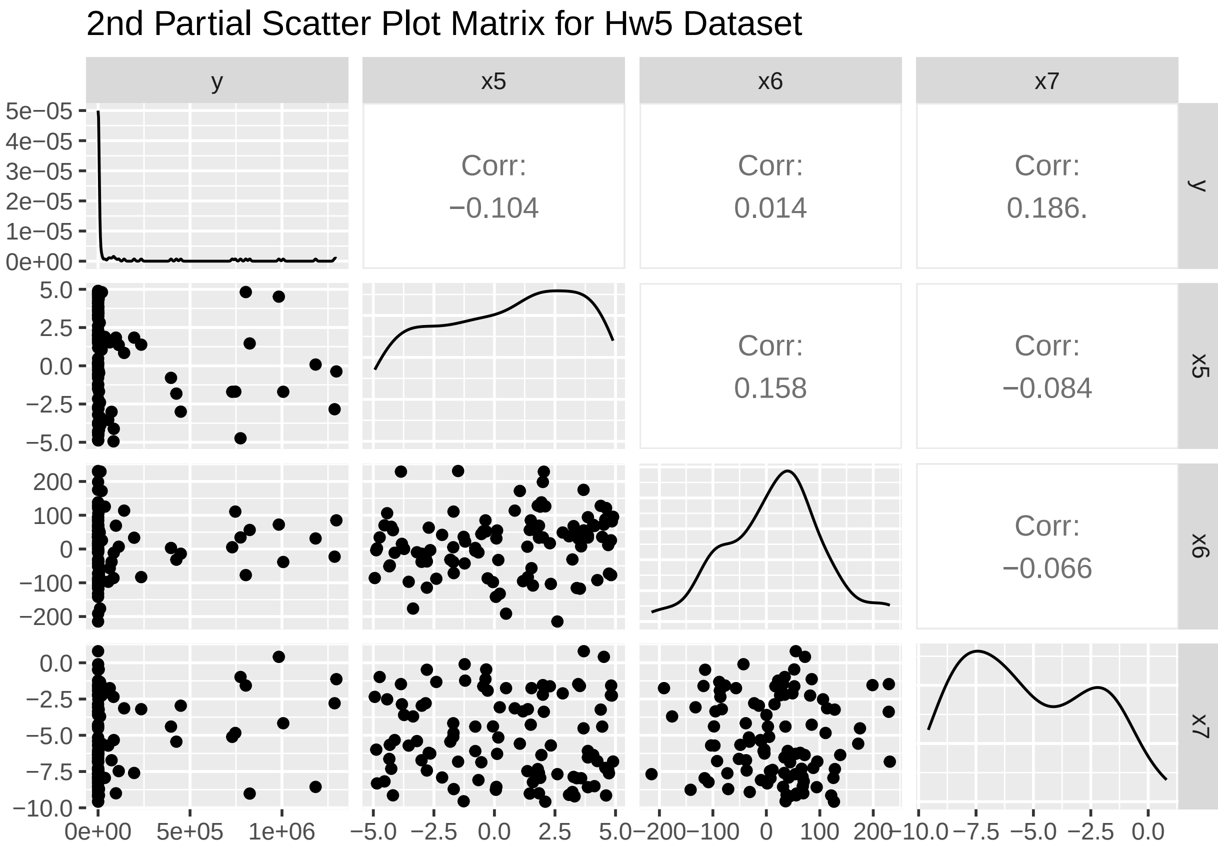
```r
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

##
## Attaching package: 'GGally'

## The following object is masked from 'package:faraway':
##
##     happy
```

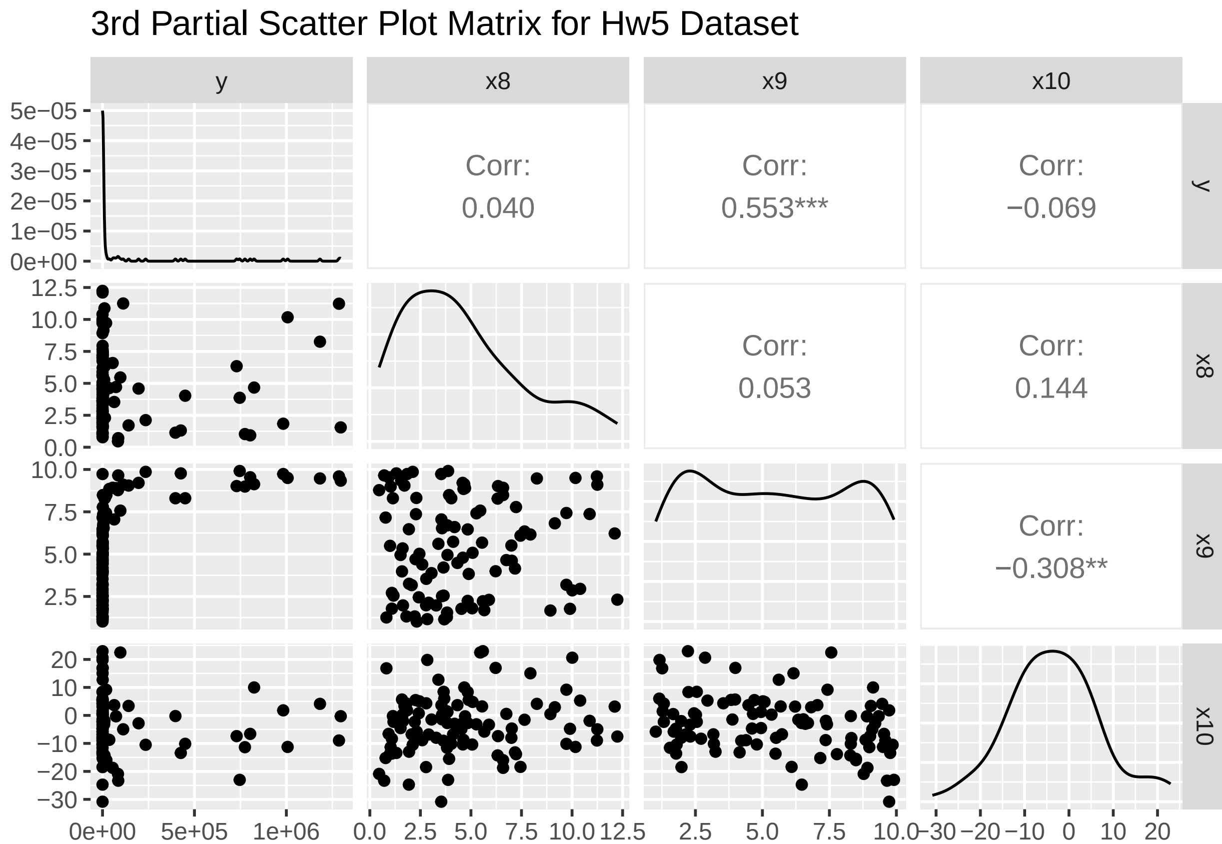
```r
ggpairs(data_hw,columns = c(1, 2:5),
                         title = "Partial Scatter Plot Matrix for Hw5 Dataset",
                         axisLabels = "show")
```




```r
ggpairs(data_hw,columns = c(1, 6:8),
                         title = "2nd Partial Scatter Plot Matrix for Hw5 Dataset",
                         axisLabels = "show")
```

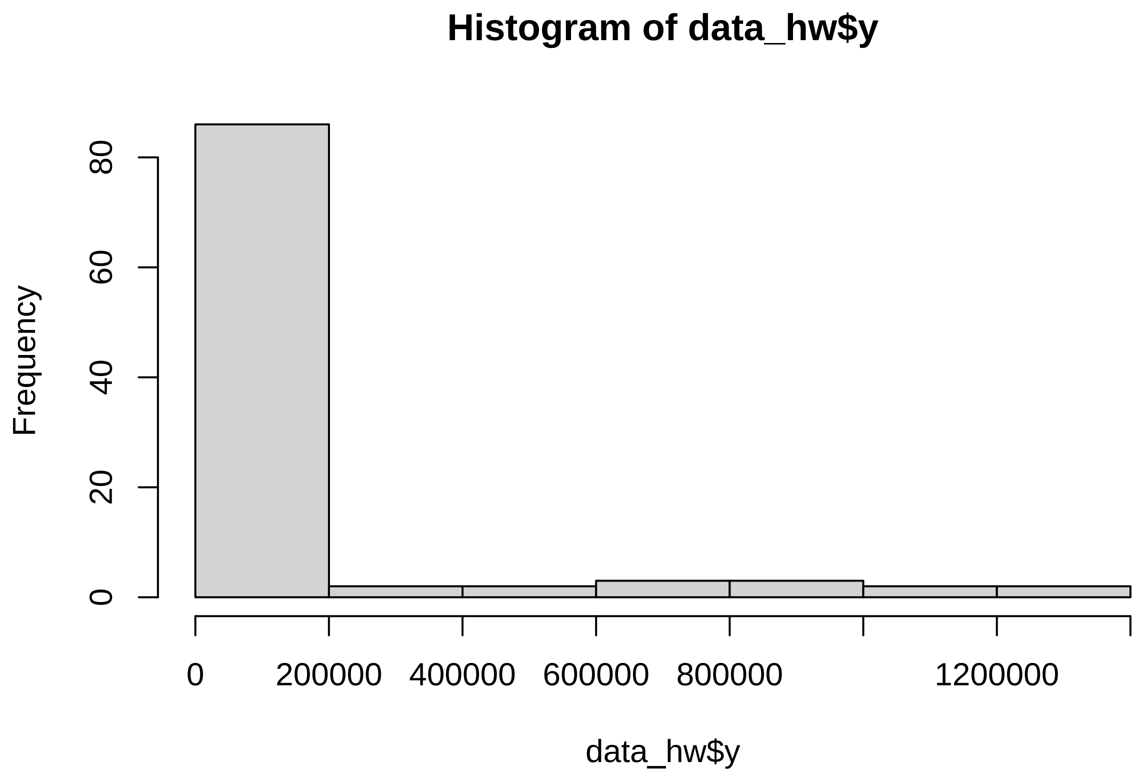



```r
ggpairs(data_hw,columns = c(1, 9:11),
                        title = "3rd Partial Scatter Plot Matrix for Hw5 Dataset",
                        axisLabels = "show")
```

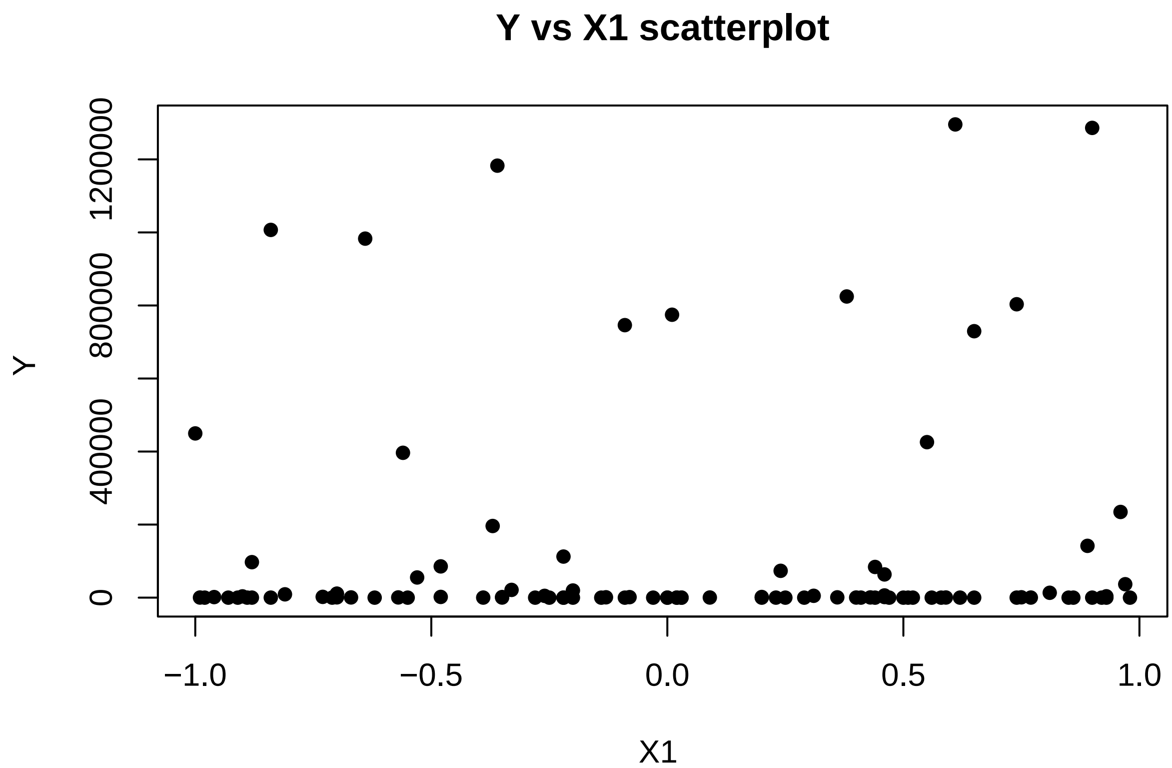


For viewing purposes, I made partial scatter plot matrices plotting y variable against each individual predictor. I notice that every single graph has the same structural problem in that most observations tend to cluster below a very low Y value and there are a few very large Y value observations. In class, we learned that in these cases transforming the response variable with a log transformation can help achieve linearity in our model.

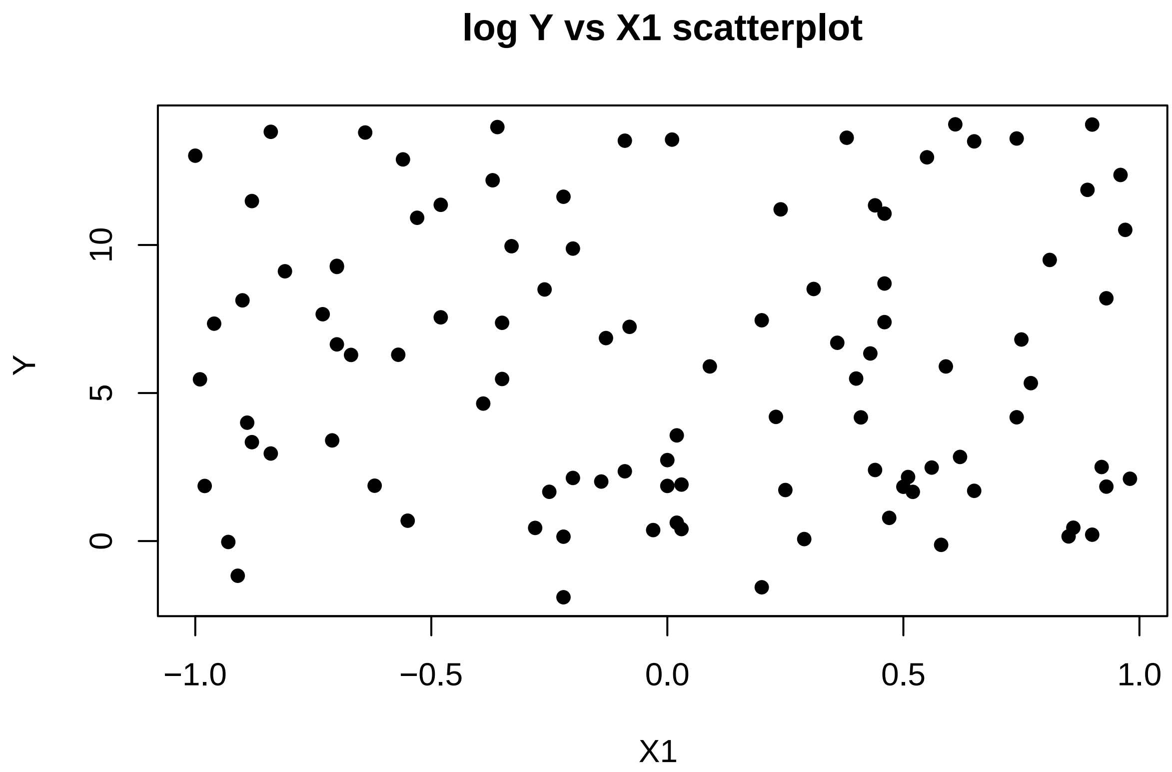
```r
##plot individual scatter plots for response variable.
hist( data_hw$y)
```



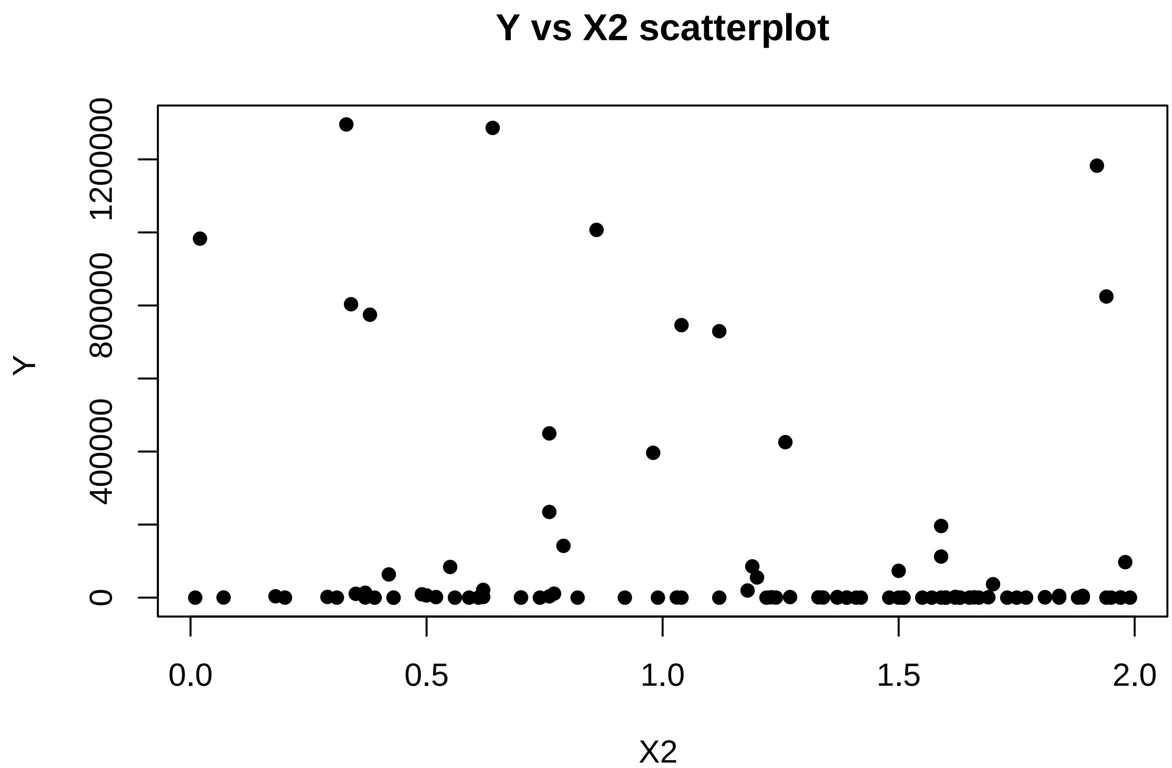

```
plot(data_hw$y ~ data_hw$x1,main ="Y vs X1 scatterplot", xlab="X1 ", ylab="Y " ,pch = 16, col = "black"
```



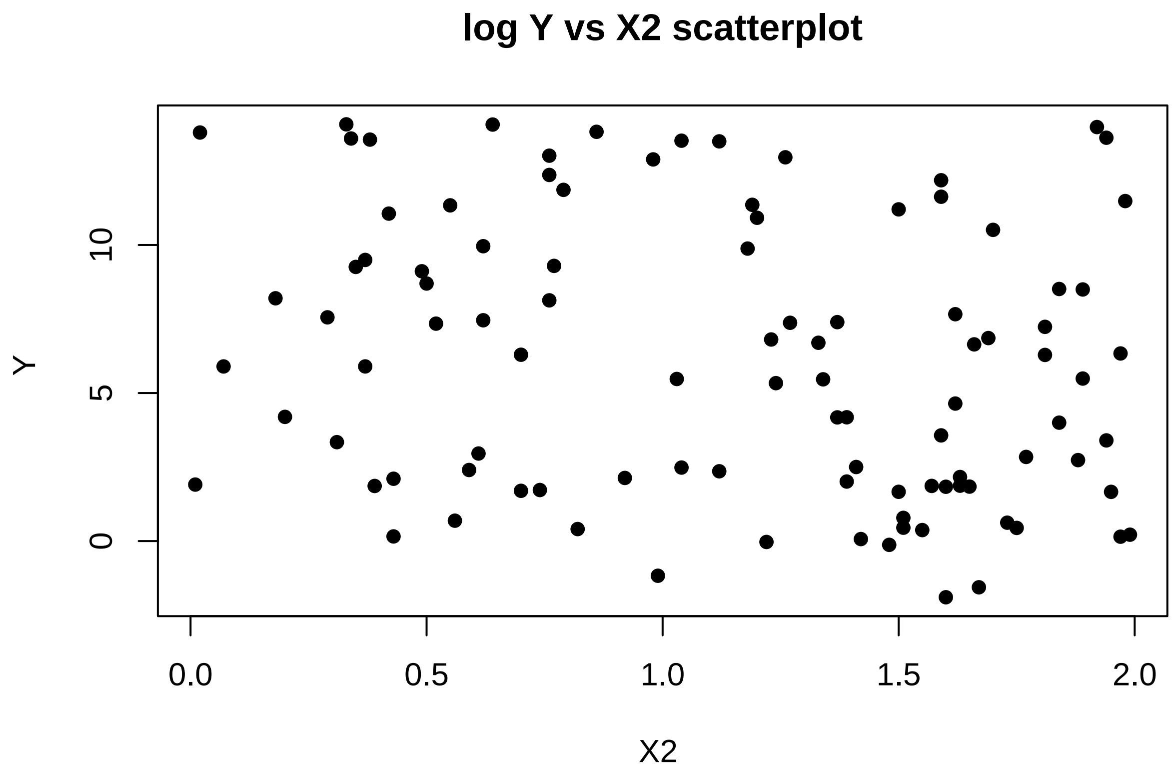

```
plot(log(data_hw$y) ~ data_hw$x1,main ="log Y vs X1 scatterplot", xlab="X1 ", ylab="Y " ,pch = 16, col =
```




```r
plot(data_hw$y ~ data_hw$x2,main ="Y vs X2 scatterplot", xlab="X2 ", ylab="Y " ,pch = 16, col = "black")
```

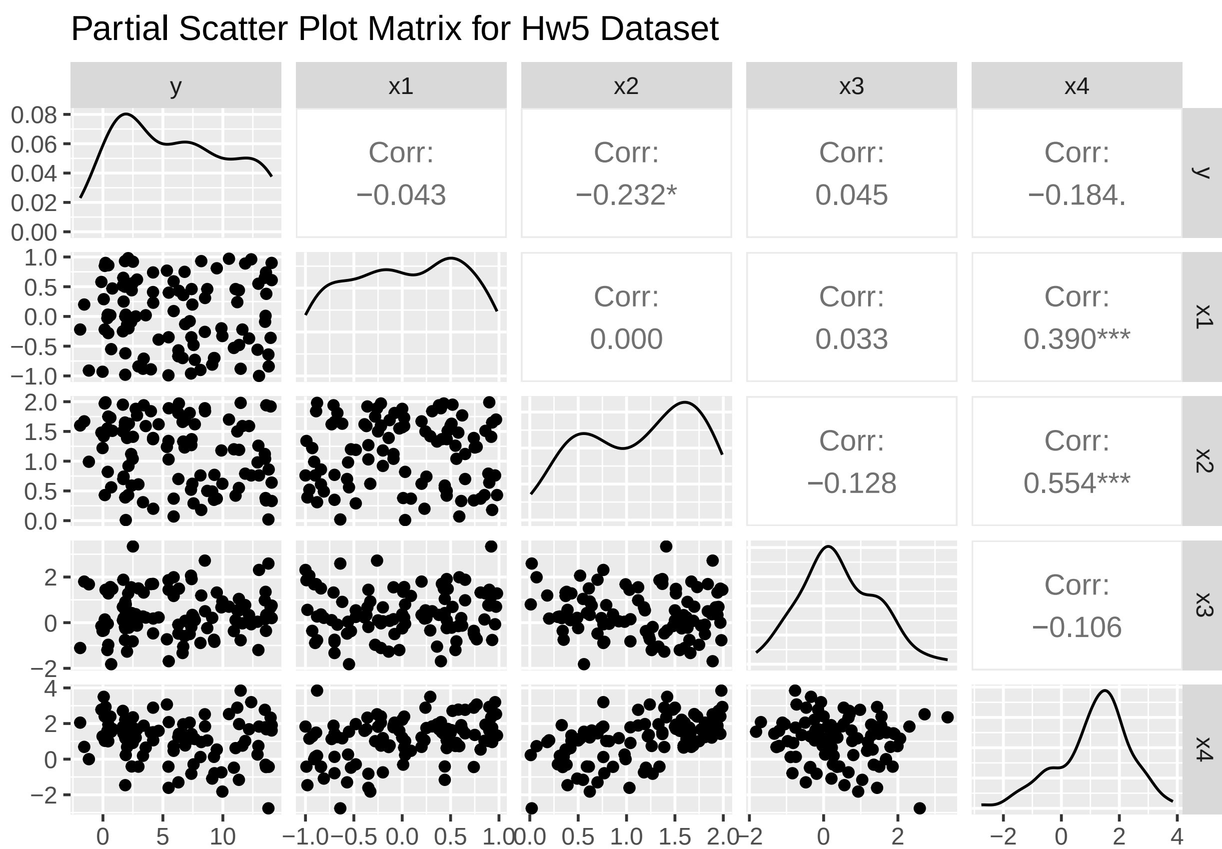



```
plot(log(data_hw$y) ~ data_hw$x2,main ="log Y vs X2 scatterplot", xlab="X2 ", ylab="Y " ,pch = 16, col =
```

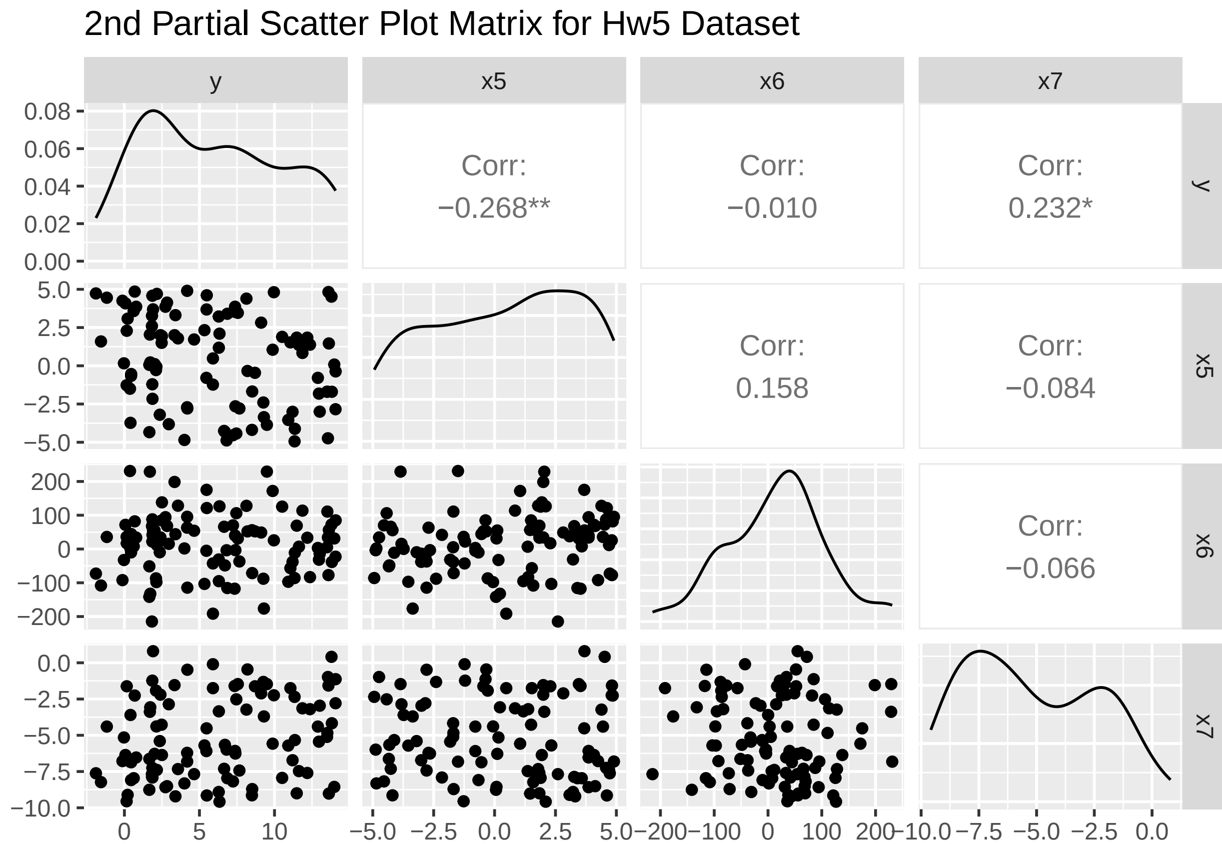


Here a histogram of Y values confirms right skew and I try plotting a log transformation on Y and graphed log Y vs a couple predictors individually. Immediately we see after transformation a much stronger linear relationship. As a result, I'm going to transform data frame with log transformation.

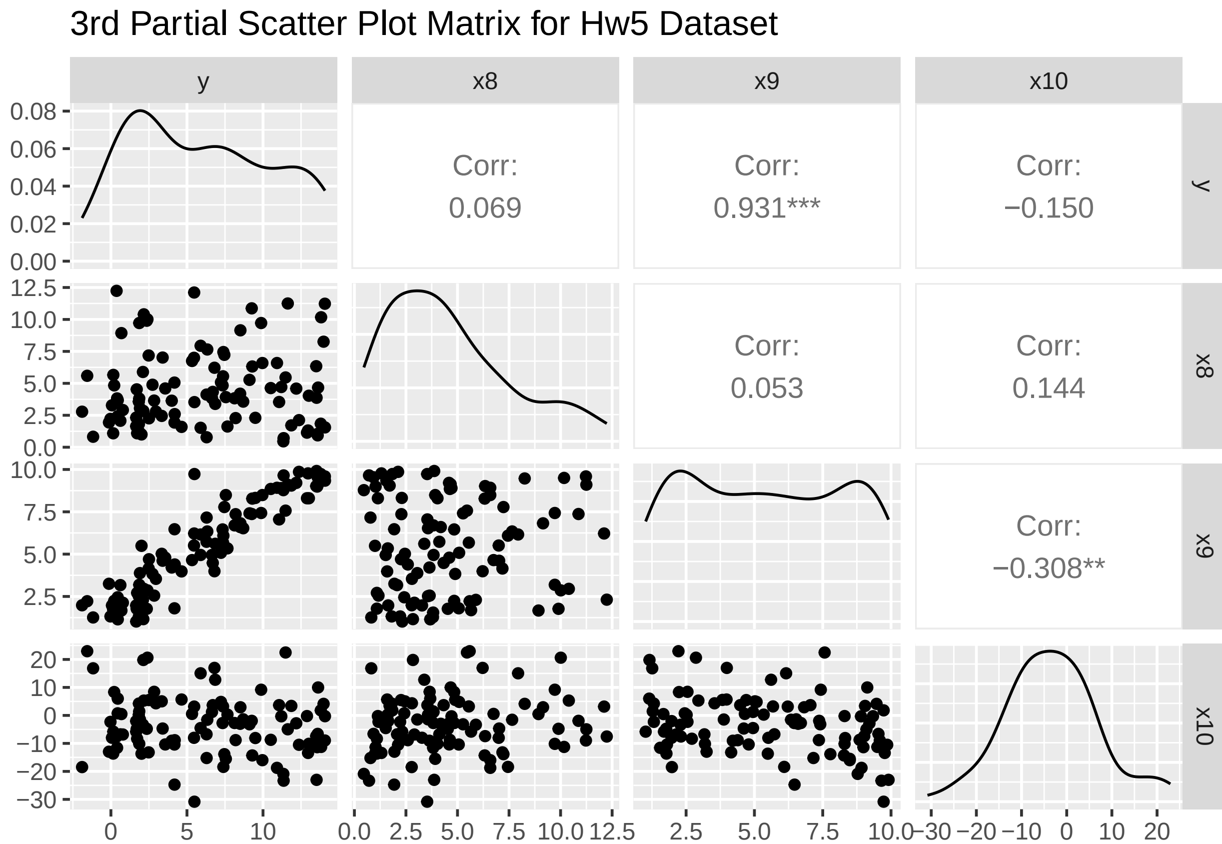
```
##create new df
df_new <- transform(data_hw, y = log(y))

 ggpairs(df_new,columns = c(1, 2:5),
                          title = "Partial Scatter Plot Matrix for Hw5 Dataset",
                          axisLabels = "show")
```




```r
ggpairs(df_new,columns = c(1, 6:8),
                         title = "2nd Partial Scatter Plot Matrix for Hw5 Dataset",
                         axisLabels = "show")
```




```
ggpairs(df_new,columns = c(1, 9:11),
                          title = "3rd Partial Scatter Plot Matrix for Hw5 Dataset",
                          axisLabels = "show")
```




```r
transform_full<- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10,data= df_new)
summary(transform_full)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9176 -0.9040  0.1104  0.8388  3.6351
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.082075   0.580134  -3.589 0.000543 ***
## x1           0.009139   0.273669   0.033 0.973436
## x2           3.158754   2.790834   1.132 0.260746
## x3          -0.003775   0.145079  -0.026 0.979298
## x4           0.109488   0.162413   0.674 0.501977
## x5          -0.229463   0.049831  -4.605 1.37e-05 ***
## x6          -0.001163   0.001647  -0.706 0.481898
## x7           0.721032   0.558582   1.291 0.200107
## x8           0.009764   0.049807   0.196 0.845024
## x9           1.559473   0.054679  28.521  < 2e-16 ***
## x10          0.076245   0.015113   5.045 2.38e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.425 on 89 degrees of freedom
## Multiple R-squared:  0.9156, Adjusted R-squared:  0.9061
## F-statistic: 96.53 on 10 and 89 DF,  p-value: < 2.2e-16
```

After log transformation, I replot the partial scatter plot matrices and the scatter plots of all individual predictors vs Y seems to have a linear relationship. Also, the summary of this new full model at least has several significant p-value coefficients. I'm now ready to attempt to build the best model from these set of predictors.

```r
library(MASS)
library(car)
backward<- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10,data= df_new)
vif(backward)
```

```
##         x1         x2         x3         x4         x5         x6         x7
##   1.318635 124.302790   1.074881   1.997535   1.113137   1.081787 120.455539
##         x8         x9        x10
##   1.069920   1.197749   1.194838
```

```r
buildBackward<-stepAIC(backward,direction="backward")
```

```
## Start:  AIC=81.24
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## - x3    1      0.00  180.84  79.243
## - x1    1      0.00  180.84  79.243
## - x8    1      0.08  180.91  79.285
## - x4    1      0.92  181.76  79.752
## - x6    1      1.01  181.85  79.801
## - x2    1      2.60  183.44  80.671
## - x7    1      3.39  184.22  81.097
## <none>               180.84  81.242
## - x5    1     43.09  223.92 100.613
## - x10   1     51.71  232.55 104.394
## - x9    1   1652.78 1833.62 310.888
##
## Step:  AIC=79.24
## y ~ x1 + x2 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## - x1    1      0.00  180.84  77.244
## - x8    1      0.08  180.92  77.287
## - x4    1      0.93  181.77  77.757
## - x6    1      1.04  181.87  77.814
## - x2    1      2.61  183.44  78.674
## - x7    1      3.39  184.23  79.099
## <none>               180.84  79.243
## - x5    1     43.87  224.71  98.965
## - x10   1     52.99  233.83 102.943
```

```
## - x9    1   1652.82 1833.66 308.890
##
## Step:  AIC=77.24
## y ~ x2 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## - x8    1      0.08  180.92  75.288
## - x6    1      1.03  181.87  75.814
## - x4    1      1.26  182.10  75.941
## - x2    1      2.62  183.46  76.685
## - x7    1      3.40  184.24  77.105
## <none>               180.84  77.244
## - x5    1     44.28  225.12  97.147
## - x10   1     53.00  233.84 100.945
## - x9    1   1653.29 1834.13 306.915
##
## Step:  AIC=75.29
## y ~ x2 + x4 + x5 + x6 + x7 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## - x6    1      0.98  181.90  73.830
## - x4    1      1.21  182.13  73.957
## - x2    1      2.62  183.54  74.724
## - x7    1      3.38  184.30  75.137
## <none>               180.92  75.288
## - x5    1     44.38  225.30  95.225
## - x10   1     55.03  235.95  99.844
## - x9    1   1671.26 1852.17 305.895
##
## Step:  AIC=73.83
## y ~ x2 + x4 + x5 + x7 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## - x4    1      1.30  183.21  72.544
## - x2    1      2.28  184.18  73.074
## - x7    1      3.01  184.92  73.474
## <none>               181.90  73.830
## - x5    1     48.12  230.02  95.299
## - x10   1     55.59  237.50  98.498
## - x9    1   1673.48 1855.39 304.068
##
## Step:  AIC=72.54
## y ~ x2 + x5 + x7 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## - x2    1      3.18  186.39  72.265
## <none>               183.21  72.544
## - x7    1      3.71  186.92  72.551
## - x5    1     49.12  232.33  94.299
## - x10   1     54.40  237.61  96.544
## - x9    1   1689.71 1872.92 303.008
##
## Step:  AIC=72.27
## y ~ x5 + x7 + x9 + x10
```

```
##
##        Df Sum of Sq     RSS     AIC
## - x7    1      2.53  188.92  71.614
## <none>               186.39  72.265
## - x5    1     52.15  238.54  94.937
## - x10   1     55.28  241.66  96.238
## - x9    1   1687.89 1874.28 301.081
##
## Step:  AIC=71.61
## y ~ x5 + x9 + x10
##
##        Df Sum of Sq     RSS     AIC
## <none>               188.92  71.614
## - x5    1     53.54  242.46  94.566
## - x10   1     54.53  243.45  94.974
## - x9    1   1771.95 1960.86 303.597
```

```
summary(buildBackward)
```

```
##
## Call:
## lm(formula = y ~ x5 + x9 + x10, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3165 -0.7945  0.1651  0.8901  3.5432
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.05063    0.30890  -6.638 1.89e-09 ***
## x5          -0.24617    0.04719  -5.216 1.05e-06 ***
## x9           1.55854    0.05194  30.007  < 2e-16 ***
## x10          0.07573    0.01439   5.264 8.59e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.403 on 96 degrees of freedom
## Multiple R-squared:  0.9118, Adjusted R-squared:  0.9091
## F-statistic: 330.9 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
vif(buildBackward)
```

```
##       x5       x9      x10
## 1.030950 1.115861 1.117788
```

First I define the full model and I check for multicollinearity. Testing variance inflation factor, we see that x2 and x7 both have VIF's above 10 which suggests multicollinearity. We don't know which predictors are correlated to x2 and x7, but our final model should likely not include these two. Next I run a stepAIC function using the backwards method to subtract predictors and to determine the model with the lowest AIC. The backward model selection gives us the model with the lowest AIC as y ~ x5+x10+x9. I also test VIF on the best model and determine there is no problem of collinearity.

```r
all <- lm(y ~ ., data=df_new)
intercept_only <- lm(y ~ 1, data=df_new)

#perform forward stepwise regression
forward <- stepAIC(intercept_only,scope=formula(all), direction='forward', )
```

```
## Start:  AIC=308.44
## y ~ 1
##
##         Df Sum of Sq     RSS    AIC
## + x9     1   1855.71  286.54 109.27
## + x5     1    153.92 1988.33 302.99
## + x7     1    115.77 2026.48 304.89
## + x2     1    115.54 2026.71 304.90
## + x4     1     72.78 2069.47 306.99
## + x10    1     48.47 2093.78 308.16
## <none>               2142.25 308.44
## + x8     1     10.34 2131.91 309.96
## + x3     1      4.43 2137.82 310.24
## + x1     1      3.90 2138.35 310.26
## + x6     1      0.20 2142.05 310.44
##
## Step:  AIC=109.27
## y ~ x9
##
##         Df Sum of Sq    RSS     AIC
## + x10    1    44.085 242.46  94.566
## + x5     1    43.094 243.45  94.974
## <none>               286.54 109.272
## + x6     1     4.718 281.83 109.612
## + x7     1     2.926 283.62 110.246
## + x3     1     2.872 283.67 110.265
## + x2     1     2.105 284.44 110.535
## + x8     1     0.855 285.69 110.974
## + x1     1     0.074 286.47 111.247
## + x4     1     0.043 286.50 111.257
##
## Step:  AIC=94.57
## y ~ x9 + x10
##
##         Df Sum of Sq    RSS    AIC
## + x5     1    53.541 188.92 71.614
## <none>               242.46 94.566
## + x6     1     4.561 237.90 94.667
## + x7     1     3.919 238.54 94.937
## + x2     1     2.999 239.46 95.322
## + x3     1     0.664 241.79 96.292
## + x4     1     0.465 241.99 96.374
## + x8     1     0.039 242.42 96.550
## + x1     1     0.000 242.46 96.566
##
## Step:  AIC=71.61
## y ~ x9 + x10 + x5
```

```
##
##        Df Sum of Sq    RSS    AIC
## <none>              188.92 71.614
## + x7    1   2.53132 186.39 72.265
## + x2    1   1.99811 186.92 72.551
## + x6    1   0.86357 188.05 73.156
## + x1    1   0.31105 188.61 73.449
## + x4    1   0.18211 188.74 73.518
## + x8    1   0.01437 188.90 73.607
## + x3    1   0.00128 188.92 73.613
```

The forward selection model suggests that Y~ X9+x10+x5 is also the best model with lowest AIC.This is promising as these two typically do not coincide exactly.

```
model1<- lm(y~x2+x5+x7+x9+x10,data= df_new)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x2 + x5 + x7 + x9 + x10, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8678 -0.8632  0.0992  0.8472  3.5440
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.03951    0.52174  -3.909 0.000175 ***
## x2           3.36785    2.63626   1.278 0.204569
## x5          -0.23731    0.04727  -5.020 2.44e-06 ***
## x7           0.73927    0.53554   1.380 0.170725
## x9           1.55235    0.05272  29.444  < 2e-16 ***
## x10          0.07572    0.01433   5.283 8.19e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.396 on 94 degrees of freedom
## Multiple R-squared:  0.9145, Adjusted R-squared:  0.9099
## F-statistic:   201 on 5 and 94 DF,  p-value: < 2.2e-16
```

```
vif(model1)
```

```
##         x2         x5         x7         x9        x10
## 115.631162   1.044205 115.429109   1.160887   1.120203
```

```
model2<- lm(y~x5+x7+x9+x10,data= df_new)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x5 + x7 + x9 + x10, data = df_new)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1935 -0.8063  0.1488  0.9388  3.5749
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.68725    0.44439  -3.797 0.000258 ***
## x5          -0.24331    0.04719  -5.156 1.37e-06 ***
## x7           0.05824    0.05128   1.136 0.258869
## x9           1.54752    0.05276  29.331  < 2e-16 ***
## x10          0.07629    0.01437   5.308 7.25e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.401 on 95 degrees of freedom
## Multiple R-squared:  0.913,  Adjusted R-squared:  0.9093
## F-statistic: 249.2 on 4 and 95 DF,  p-value: < 2.2e-16
```

```
vif(model2)
```

```
##       x5       x7       x9      x10
## 1.033901 1.051276 1.154920 1.119112
```

```
model_best<- lm(y~x5+x9+x10,data= df_new)
summary(model_best)
```

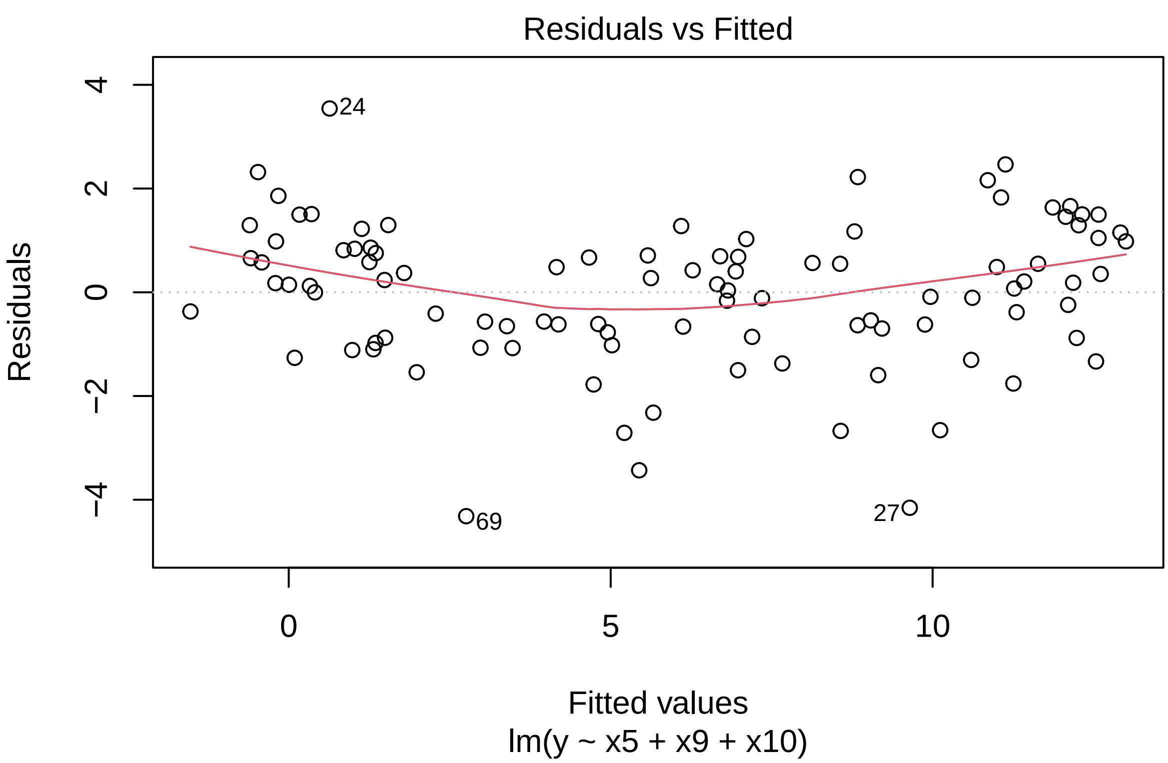
```
##
## Call:
## lm(formula = y ~ x5 + x9 + x10, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3165 -0.7945  0.1651  0.8901  3.5432
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.05063    0.30890  -6.638 1.89e-09 ***
## x5          -0.24617    0.04719  -5.216 1.05e-06 ***
## x9           1.55854    0.05194  30.007  < 2e-16 ***
## x10          0.07573    0.01439   5.264 8.59e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.403 on 96 degrees of freedom
## Multiple R-squared:  0.9118, Adjusted R-squared:  0.9091
## F-statistic: 330.9 on 3 and 96 DF,  p-value: < 2.2e-16
```

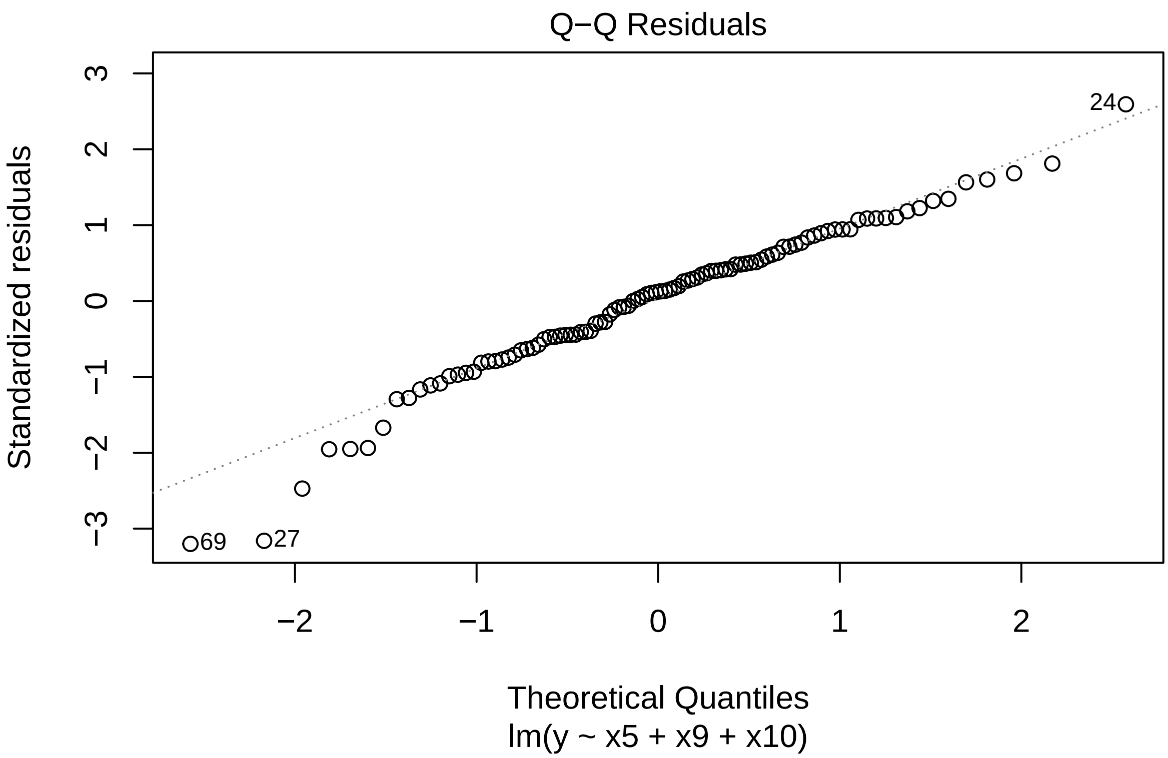
```
vif(model_best)
```

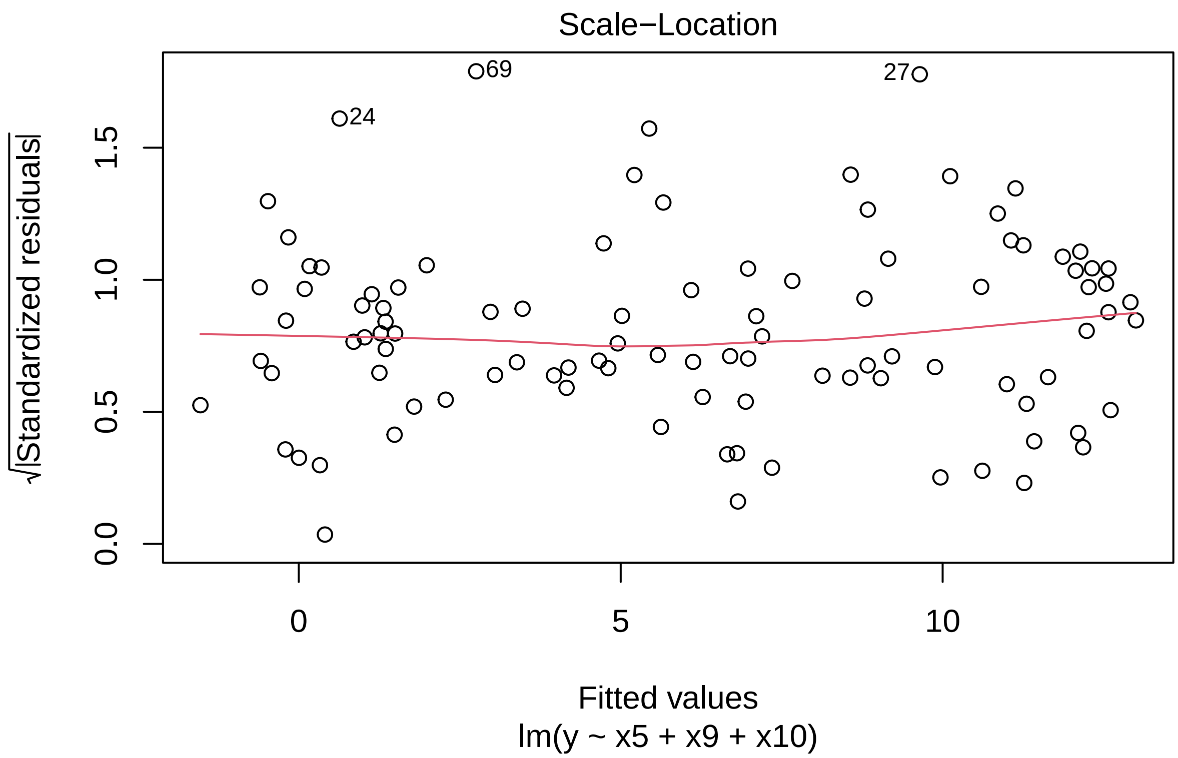
```
##       x5       x9      x10
## 1.030950 1.115861 1.117788
```

Out of curiousity I'm comparing the R summary output of the last 3 models provided by the StepAIC process and comparing their adjusted R^2. Model 1 has problem of multi-collinearity still has the VIFS of the remaining 5 predictors still have x2 and x7 as high VIF's. Therefore it makes sense to take one more backward step. Model 2 has no problem with collinearity based on VIFS. However, x7 still isn't a significant predictor. Comparing that to the best model, we see that the best model has the remaining 3 predictors all as significant. Additionally, I sacrifice very little adjusted R^2 going from model 2 to the best model and by principle or parismony I suspect that the model provided by the Step AIC is the best model.

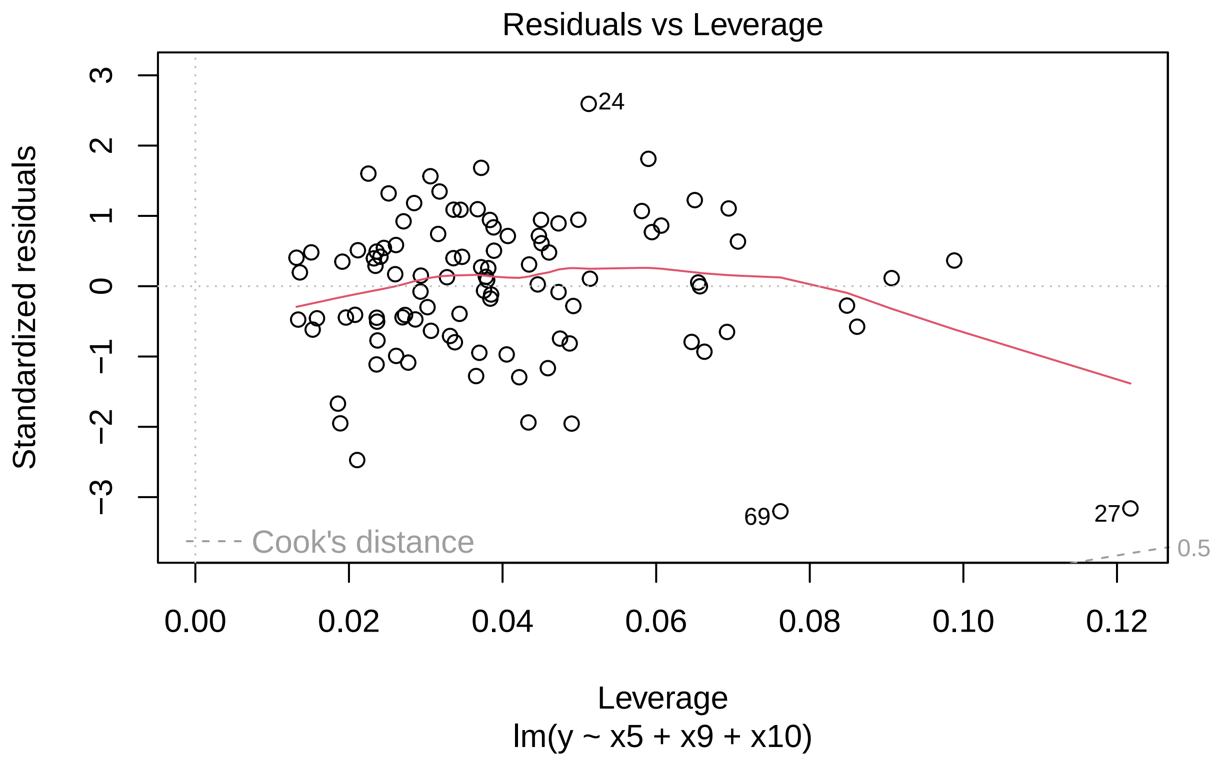
```
plot(model_best)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(y ~ x5 + x9 + x10)
−3
−2
−1
0
1
2
3
69
27
24



Scale−Location
69
24
27
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
5
10
Fitted values
lm(y ~ x5 + x9 + x10)



Checking the assumptions for regression for the best model, I see that issues with homoskedacity are fixed. The residuals seem scattered with non-constant variance. Aside from a couple potential outliers, the normality assumption from QQ plot seems met. None of the outlying points seem to have too large influence based on Residuals vs leverage plot and cook's distance.

```
best_mod_interact<- lm(y ~ (x5+x9+x10)^2,data=df_new)
summary(best_mod_interact)
```

```
##
## Call:
## lm(formula = y ~ (x5 + x9 + x10)^2, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5926 -0.7838  0.1610  0.8047  2.9189
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.222432   0.315264  -7.049  3.1e-10 ***
## x5          -0.265786   0.102824  -2.585   0.0113 *
## x9           1.600073   0.055574  28.792  < 2e-16 ***
## x10          0.003892   0.030243   0.129   0.8979
## x5:x9        0.009814   0.018397   0.533   0.5950
## x5:x10       0.006608   0.004850   1.363   0.1763
## x9:x10       0.011610   0.004667   2.487   0.0146 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.368 on 93 degrees of freedom
## Multiple R-squared:  0.9187, Adjusted R-squared:  0.9135
## F-statistic: 175.2 on 6 and 93 DF,  p-value: < 2.2e-16
```

```r
best_mod_interact1<-lm(y~x5+x9+x10+x9:x10, data=df_new)
summary(best_mod_interact1)
```

```
##
## Call:
## lm(formula = y ~ x5 + x9 + x10 + x9:x10, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5499 -0.7840  0.1470  0.7131  3.1696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.219941   0.308799  -7.189 1.48e-10 ***
## x5          -0.249538   0.046016  -5.423 4.45e-07 ***
## x9           1.607396   0.054365  29.567  < 2e-16 ***
## x10          0.012654   0.029188   0.434   0.6656
## x9:x10       0.011228   0.004557   2.464   0.0155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.367 on 95 degrees of freedom
## Multiple R-squared:  0.9171, Adjusted R-squared:  0.9136
## F-statistic: 262.8 on 4 and 95 DF,  p-value: < 2.2e-16
```

```r
best_mod_interact2<-lm(y~x5+x9+x9:x10, data=df_new)
summary(best_mod_interact2)
```

```
##
## Call:
## lm(formula = y ~ x5 + x9 + x9:x10, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5980 -0.8247  0.1336  0.7085  3.0831
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.23924    0.30428  -7.359 6.29e-11 ***
## x5          -0.24903    0.04581  -5.437 4.12e-07 ***
## x9           1.61183    0.05317  30.317  < 2e-16 ***
## x9:x10       0.01296    0.00218   5.946 4.43e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.361 on 96 degrees of freedom
## Multiple R-squared:  0.9169, Adjusted R-squared:  0.9144
## F-statistic: 353.3 on 3 and 96 DF,  p-value: < 2.2e-16
```

Finally, I would like to check for interaction effects among the remaining 3 predictors I'm including in the model. First I run a model with all 2 way interaction terms present between remaining predictors. First model only the x9:x10 interaction term is significant. I run a second model with all original predictors + the x9:x10 interaction term. The adjusted R^2 of model improves, but it now appears that x10 predictor is no longer significant. I also run a 3rd model by dropping the main effect of x10. This model has even better adjusted R^2, has fewer overall predictors, and has all statistically significant p-value predictors.

In conclusion, I would argue that either the model y~ x5 + x9 + x10 + x9:x10 or y ~ x5 + x9 + x9:x10 are the best model available. First, I log transformed the y values of the dataset to obtain a more linear relationship between response and predictors. I then used backward step AIC model selection to determine which set of predictors gave the model with least AIC. I checked VIF for the full model and determined high collinearity for x2 and x7 which were eventually excluded from best model. The best model has no meaningful issue with collinearity. I validated, that my best model met all linear regression assumptions. Finally, I attempted inclusion of interaction effect which did seem to produce even better adjusted R^2 values. However, given that all predictor variables are continous and that this is an educational dataset, I have no physical context to really interpret what the interaction effect means, even if it appears to improve the model. Although, it is feasible to drop one of the main effects (x9+x10) of the x9:x10 interaction term, I'm hesitant to do so without additional context. In either case, based on these procedures, validating my model for interaction terms, transformations, and usage of selection procedure, I believe these two model above are the best possible linear model.