Keiland Pullen
ID:1977120

DSC 423 | Spring 2021
Assignment 4

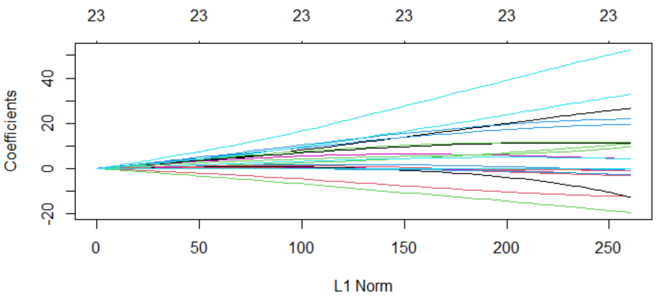
Honor statement:
"I have completed this work independently. The solutions given are entirely my own work"

1a.)
The following is the code used to perform Ridge Regression on the Piso2009 dataset:

```
View(Pisa2009)
library(glmnet)

# Get all rows and columns except for the 1st and 25th columns.
x <- data.matrix(Pisa2009[,2:24])

# Get the response variable and save it as a double data-type.
y <- as.double(Pisa2009[,25])

# Create the model w/ alpha = 0 for Ridge.
pisa_model <- glmnet(x, y, alpha = 0)
summary(pisa_model)
```

```
> summary(pisa_model)

                Length      Class           Mode
a0              100         -none-          numeric
beta            2300        dgCMatrix S4
df              100         -none-          numeric
dim             2           -none-          numeric
lambda          100         -none-          numeric
dev.ratio       100         -none-          numeric
nulldev         1           -none-          numeric
npasses         1           -none-          numeric
jerr            1           -none-          numeric
offset          1           -none-          logical
call            5           -none-          call
nobs            1           -none-          numeric
```

```
# Displays Trace Plot
plot(pisa_model)
```



```
# cv automatically does cross-validation to identify the lambda
pisa_ridge_cv <- cv.glmnet(x,y, family="gaussian", alpha=0)
pisa_ridge_cv
```

```
> pisa_ridge_cv

Call:  cv.glmnet(x = x, y = y, family = "gaussian", alpha = 0)

Measure: Mean-Squared Error

        Lambda     Index  Measure   SE Nonzero
min      3.36      99     5705    128.1    23
1se     41.41      72     5819    136.7    23
```

```
# Get the lambda
pisa_lambda <- pisa_ridge_cv$lambda.min
pisa_lambda
```

```
> pisa_lambda <- pisa_ridge_cv$lambda.min
> pisa_lambda
[1] 3.359216
```

```
# Find coefficients of the model
coef(pisa_ridge_cv, s=pisa_lambda)
```

```
> coef(pisa_ridge_cv, s=pisa_lambda)
24 x 1 sparse Matrix of class "dgCMatrix"
                                   1
(Intercept)              178.907609883
grade                     26.561537211
male                     -12.406794130
raceeth                   10.999647245
preschool                 -0.740149794
expectBachelors           52.282541085
motherHS                   4.342749265
motherBachelors           11.154201099
motherWork                -3.198076587
fatherHS                  11.604885058
fatherBachelors           19.515312833
fatherWork                 4.246623659
selfBornUS                 0.134092464
motherBornUS             -12.584452833
fatherBornUS              -2.535264505
englishAtHome              9.588211699
computerForSchoolwork     21.916035046
read30MinsADay            32.661212423
minutesPerWeekEnglish      0.014312649
studentsInEnglish         -0.027115779
schoolHasLibrary          -1.045897572
publicSchool             -19.436026300
urban                     -2.768863426
schoolSize                 0.006535571
```

The following is the single-order model that was created with the coefficients of the Ridge Regression technique:

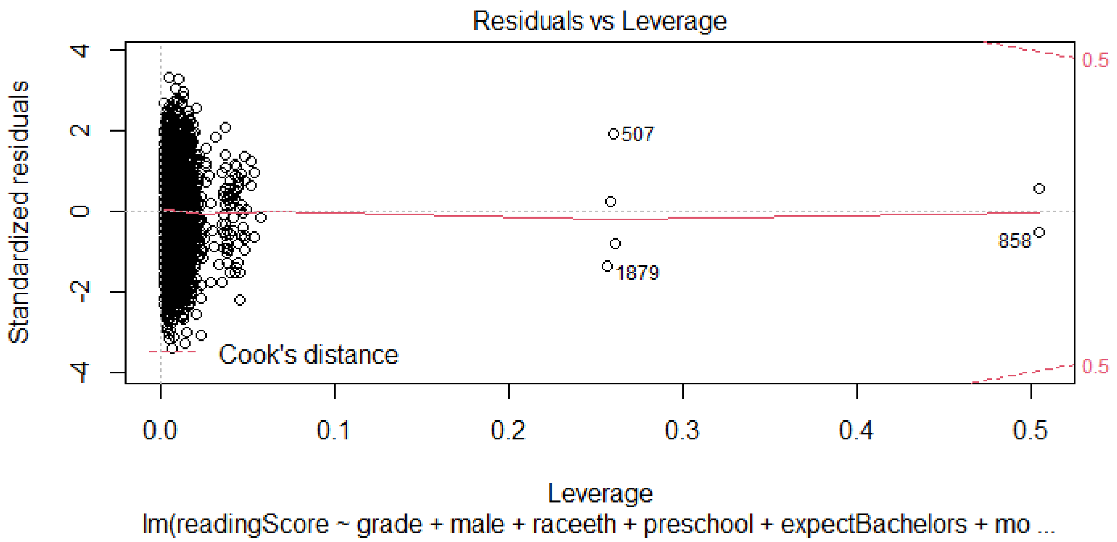
```
> pisa_model_1 <- lm(readingScore ~ grade + male + raceeth + preschool + expectBachelors + motherHS
+                  + motherBachelors + motherWork + fatherHS + fatherBachelors + fatherWork + selfBornUS
+                  + motherBornUS + fatherBornUS + englishAtHome + computerForSchoolwork + read30MinsADay
+                  + minutesPerWeekEnglish + studentsInEnglish + schoolHasLibrary + publicSchool
+                  + urban + schoolSize, data = Pisa2009)
> summary(pisa_model_1)

Call:
lm(formula = readingScore ~ grade + male + raceeth + preschool +
    expectBachelors + motherHS + motherBachelors + motherWork +
    fatherHS + fatherBachelors + fatherWork + selfBornUS + motherBornUS +
    fatherBornUS + englishAtHome + computerForSchoolwork + read30MinsADay +
    minutesPerWeekEnglish + studentsInEnglish + schoolHasLibrary +
    publicSchool + urban + schoolSize, data = Pisa2009)

Residuals:
     Min       1Q   Median       3Q      Max
-252.698  -48.479    0.481   49.936  247.243

Coefficients:
                                            Estimate     Std. Error    t value   Pr(>|t|)
(Intercept)                                 299.488737   55.703871     5.376     8.11e-08 ***
grade9                                      40.285804    52.722943     0.764     0.444859
grade10                                     90.414303    52.582799     1.719     0.085621 .
grade11                                     104.977136   52.650018     1.994     0.046247 *
grade12                                     153.124134    64.477284    2.375     0.017612 *
male1                                       -12.629264   2.644476      -4.776    1.87e-06 ***
raceethAsian                                59.289097    15.422137     3.844     0.000123 ***
raceethBlack                                -3.245780    14.086553     -0.230    0.817782
raceethHispanic                             28.478156    13.967874     2.039     0.041545 *
raceethMore than one race                   42.834427    15.091587     2.838     0.004563 **
raceethNative Hawaiian/Other Pacific Islander  52.643342    20.069600  2.623     0.008754 **
raceethWhite                                62.865269    13.555355     4.638     3.66e-06 ***
preschool1                                  -2.008505    2.956941      -0.679    0.497026
expectBachelors1                            53.227613    3.576399      14.883    < 2e-16 ***
motherHS1                                   4.375418     5.063943      0.864     0.387631
motherBachelors1                            11.151077    3.281944      3.398     0.000687 ***
motherWork1                                 -2.268512    2.953436      -0.768    0.442486
fatherHS1                                   6.891077     4.667189      1.476     0.139905
fatherBachelors1                            17.604801    3.384319      5.202     2.09e-07 ***
fatherWork1                                 3.033776     3.695971      0.821     0.411799
selfBornUS1                                 0.796308     5.976750      0.133     0.894016
motherBornUS1                               -8.337533    5.669328      -1.471    0.141482
fatherBornUS1                                2.556788    5.369816      0.476     0.634005
englishAtHome1                              10.905428    5.835964      1.869     0.061757 .
computerForSchoolwork1                      19.807206    4.856144      4.079     4.63e-05 ***
read30MinsADay1                             32.736380    2.862328      11.437    < 2e-16 ***
minutesPerWeekEnglish                       0.011938     0.009016      1.324     0.185567
studentsInEnglish                           -0.182103    0.192915      -0.944    0.345260
schoolHasLibrary1                           -1.019002    7.570382      -0.135    0.892933
publicSchool1                               -18.794210   5.590012      -3.362    0.000782 ***
urban1                                      -1.563926    3.320545      -0.471    0.637682
schoolSize                                  0.007573     0.001813      4.177     3.03e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 74.07 on 3372 degrees of freedom
Multiple R-squared:  0.3161,           Adjusted R-squared:  0.3099
F-statistic: 50.29 on 31 and 3372 DF,  p-value: < 2.2e-16
```

The following were continued to check the Residual Standard Error and the Adjusted R-Square value that was calculated in the above model:

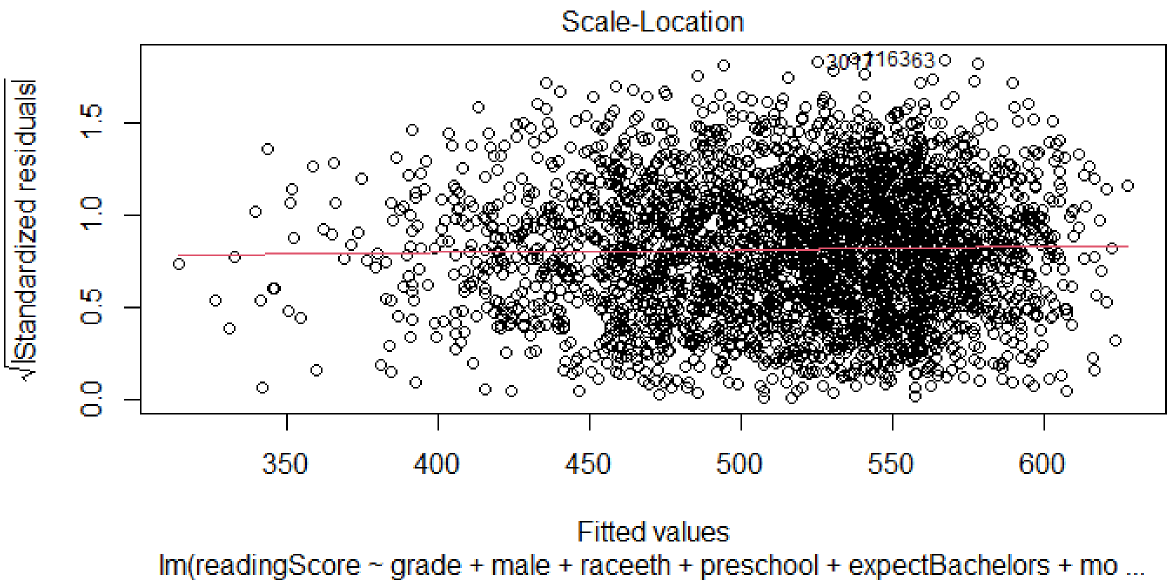
```
> y_predicted <- predict(pisa_ridge_cv, s = pisa_lambda, newx = x)
> sst <- sum((y - mean(y))^2)
> sse <- sum((y_predicted - y)^2)
> rsq <- 1 - sse/sst
> rsq
[1] 0.2925447
> RMSE = sqrt(sse/nrow(Pisa2009))
> RMSE
[1] 74.98547
```

The Ridge Trace Plot listed earlier demonstrates how Ridge Regression adds a degree of bias to the estimates and reduces the standard errors. The idea is for this technique to provide more reliable estimates.

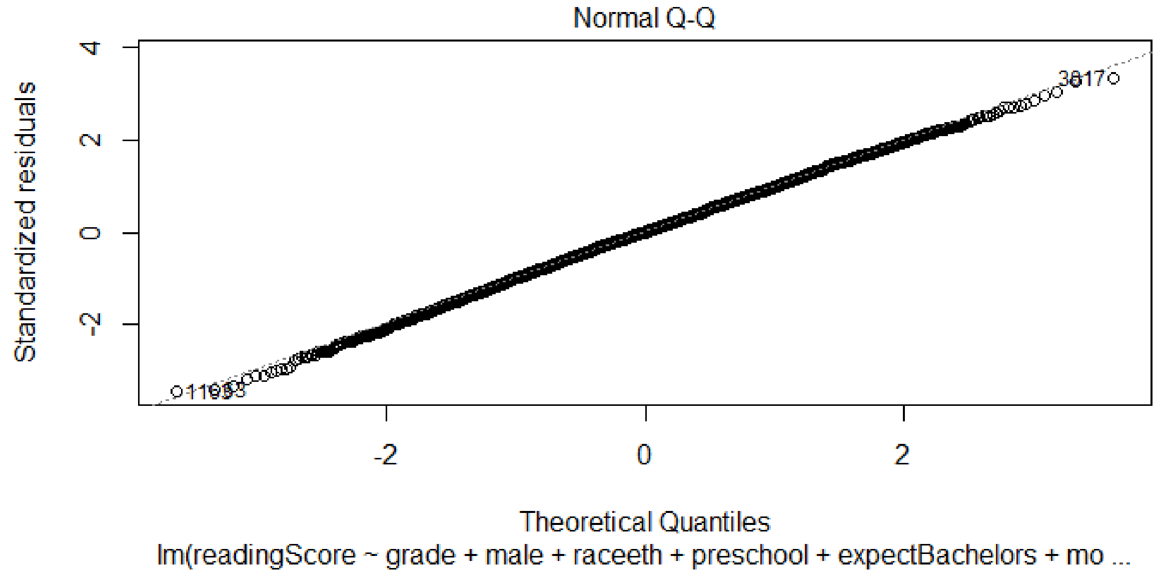
The residual plots are as follows:



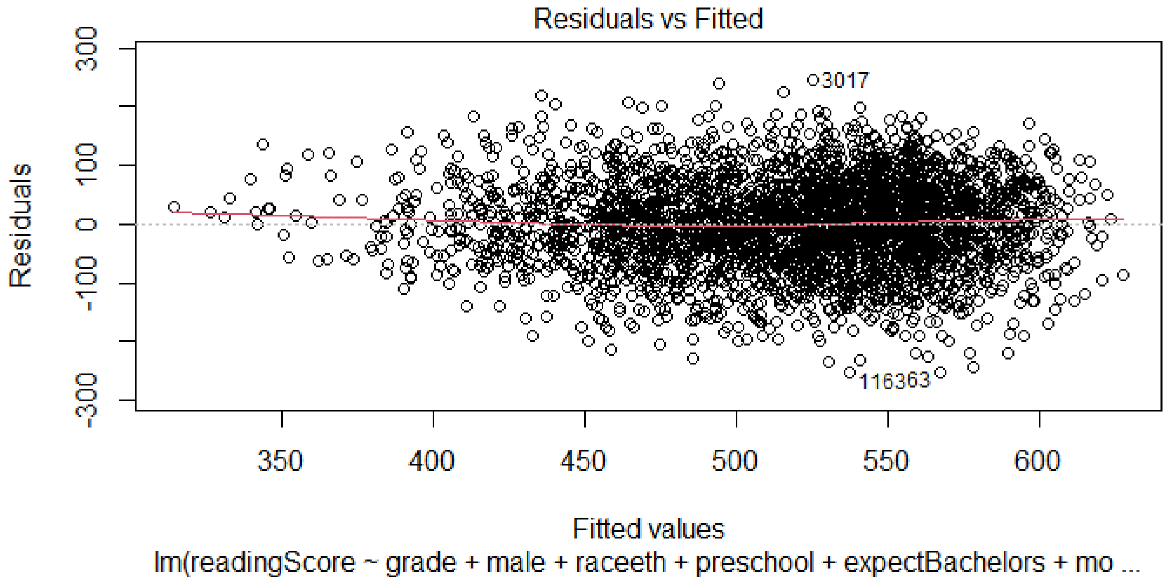
The “Residuals vs Leverage” plot will assist us with identifying outliers which may have influence on the change in the slope of the line.



The Scale-Location Plot above verifies the homoskedasticity assumption.



The above QQ Plot displays that a majority of the data are on the line, which indicates that the residuals are normal.



The above “Residuals vs Fitted” Plot displays heteroscedacity as there is different variance between that data points at the left-end and right-end of the plot. I suppose that there could be some argument for homoscedacity as well. This is something that I’ll need to investigate further.

1.b)

The following commands were used to generate the LASSO coefficients:

```
set.seed(123) # random numbers
pisa_lasso <- cv.glmnet(x,y, family="gaussian", alpha=1)
coef(pisa_lasso,  s=pisa_lambda)
```

The coefficients are as follows:

```
> set.seed(123) # random numbers
> pisa_lasso <- cv.glmnet(x,y, family="gaussian", alpha=1)
> coef(pisa_lasso,  s=pisa_lambda)
24 x 1 sparse Matrix of class "dgCMatrix"
                                   1
(Intercept)                        196.693536580
grade                              23.868517417
male                               -6.181739917
raceeth                            8.823500814
preschool                          .
expectBachelors                    51.560027629
motherHS                           .
motherBachelors                    7.896835697
motherWork                         .
fatherHS                           5.882282620
fatherBachelors                    20.590565216
fatherWork                         .
selfBornUS                         .
motherBornUS                        .
fatherBornUS                       .
englishAtHome                      .
computerForSchoolwork              17.527065546
read30MinsADay                     28.089591247
minutesPerWeekEnglish              .
studentsInEnglish                  .
schoolHasLibrary                    .
publicSchool                       -4.643123913
urban                              .
schoolSize                         0.001126722
```

The following is the model created that uses the features generated via LASSO:

```
Call:
lm(formula = readingScore ~ grade + male + raceeth + expectBachelors +
    motherBachelors + fatherHS + fatherBachelors + computerForSchoolwork +
    read30MinsADay + publicSchool + schoolSize, data = Pisa2009)

Residuals:
     Min       1Q   Median       3Q      Max
-256.107  -48.660    1.594   49.687  244.674

Coefficients:
                                              Estimate     Std. Error    t value  Pr(>|t|)
(Intercept)                                   304.822253   54.450077     5.598    2.34e-08 ***
grade9                                        38.727433    52.641785     0.736    0.461977
grade10                                       88.635051    52.493357     1.689    0.091407 .
grade11                                       103.518161   52.557953     1.970    0.048966 *
grade12                                       155.501148   64.278458     2.419    0.015608 *
male1                                         -12.442632   2.634792      -4.722   2.43e-06 ***
raceethAsian                                  58.293463    14.887589     3.916    9.20e-05 ***
raceethBlack                                  -3.686041    14.056466     -0.262   0.793159
raceethHispanic                               26.763415    13.724340     1.950    0.051250 .
raceethMore than one race                     43.606069    15.051915     2.897    0.003791 **
raceethNative Hawaiian/Other Pacific Islander 54.683162    19.837983     2.756    0.005874 **
raceethWhite                                  62.922968    13.517890     4.655    3.37e-06 ***
expectBachelors1                              53.096680    3.559075      14.919   < 2e-16 ***
motherBachelors1                              11.212567    3.233992      3.467    0.000533 ***
fatherHS1                                     9.574209     4.179756      2.291    0.022047 *
fatherBachelors1                              18.147346    3.355885      5.408    6.83e-08 ***
computerForSchoolwork1                        20.322678    4.797978      4.236    2.34e-05 ***
read30MinsADay1                               32.886562    2.851702      11.532   < 2e-16 ***
publicSchool1                                 -17.583400   4.962743      -3.543   0.000401 ***
schoolSize                                    0.006759     0.001623      4.165    3.19e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 74.05 on 3384 degrees of freedom
Multiple R-squared:  0.3141,      Adjusted R-squared:  0.3102
F-statistic: 81.55 on 19 and 3384 DF,  p-value: < 2.2e-16
```

1c.)

In this particular case, the models are not identical, but they are very close as the Adj. R-Squared value of the model based on Ridge is 30.99% and the Adj. R-Squared value for the model based on LASSO is 31%. Another difference is how the model variables are created. Ridge creates all of the variables and they are then added to model. For LASSO, the variables that are not initially needed are zeroed out. This demonstrates how useful LASSO regression is with feature selection.

2a. & 2b.)

The following code was used to create the Logistic model:

```
# Convert to factor because the current variable contains two-levels
remission$remiss <- as.factor(remission$remiss)
summary(remission)

# Using "family = binomial" to tell us that we're using Logistic Regression
rem_model <- glm(remiss ~., family = "binomial", data=remission)
summary(rem_model)
```

```
Call:
glm(formula = remiss ~ ., family = "binomial", data = remission)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.95165  -0.66491  -0.04372   0.74304   1.67069

Coefficients:
             Estimate    Std. Error    z value  Pr(>|z|)
(Intercept)  58.0385     71.2364       0.815    0.4152
cell         24.6615     47.8377       0.516    0.6062
smear        19.2936     57.9500       0.333    0.7392
infil        -19.6013    61.6815       -0.318   0.7507
li           3.8960      2.3371        1.667    0.0955 .
blast        0.1511      2.2786        0.066    0.9471
temp         -87.4339    67.5735       -1.294   0.1957
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.372  on 26  degrees of freedom
Residual deviance: 21.751  on 20  degrees of freedom
AIC: 35.751

Number of Fisher Scoring iterations: 8
```

2c.)

The glm() function means “General Linear Model”. The lm() function fits models in the form of Y =Xb + e, where glm() fits models in the form of f(Y) = Xb + e and the “e” or distribution of the error term can be specified.

2d.)

Using the glm() function doesn’t appear to create a model that fits the dataset “remission”. The p-values for each of the variables aren’t good.

The following will display the confidence interval and coefficients of the model, however, since this model appears to be a bad fit, I'm unsure if we should proceed with these steps:

```
# Confidence interval of the model at 95%
confint(rem_model)

# Coefficients of the model
coef(rem_model)

# This "delogs" the coefficient
exp(coef(rem_model)) -1
```

```
> confint(rem_model)  # Confidence interval of the model.. at 95%
                     2.5 %       97.5 %
(Intercept)      -70.9683777   222.202990
cell             -27.7332544   138.404531
smear            -60.4544868   152.174139
infil            -159.7565104  67.536927
li               0.1944541     9.526820
blast            -4.5238625    4.715064
temp             -244.7720744  24.913187
There were 26 warnings (use warnings() to see them)

> coef(rem_model) # Coefficients of the model
(Intercept)       cell          smear         infil          li          blast           temp
 58.0384871    24.6615439  19.2935746 -19.6012612   3.8959633   0.1510923 -87.4339024
>
>
> exp(coef(rem_model)) -1  # This "delogs" the coefficient..
  (Intercept)            cell            smear           infil             li                blast
 1.606182e+25  5.133014e+10  2.393828e+08 -1.000000e+00  4.820343e+01  1.631040e-01

        temp
-1.000000e+00
```