INTRODUCTION

This report includes a diagnosis that shows how to use RStudio to fit, analyses, and assess a regression model. This assignment module's learning outcomes are as follows:

• Conduct regularization method for models to describe relationships among variables and make useful predictions.

What is Lasso Regression?

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e., models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination (S., 2021).

The acronym "LASSO" stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator.

L1 Regularization

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g., Ridge regression) *doesn't* result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge (S., 2021).

What is Ridge Regression?

Ridge regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity (correlations between predictor variables). Ridge regression uses a type of shrinkage estimator called a *ridge estimator*. Shrinkage estimators theoretically produce new estimators that are shrunk closer to the "true" population parameters. The ridge estimator is especially good at improving the least-squares estimate when multicollinearity is present.

L2 Regularization

Ridge regression belongs a class of regression tools that use L2 regularization. The other type of regularization, **L1 regularization**, limits the size of the coefficients by adding an *L1 penalty* equal to the absolute value of the magnitude of coefficients. This sometimes results in the elimination of some coefficients altogether, which can yield sparse models. **L2 regularization** adds an L2 penalty, which equals the square of the magnitude of coefficients. All coefficients are shrunk by the same factor (so none are eliminated). Unlike L1 regularization, L2 will *not* result in sparse models (S., 2021).

ANALYSIS SECTION

Libraries Used

```
#Libraries used

library(ISLR)
library(ggplot2)
library(dlookr)
library(caret)
library(DT)
library(pROC)
library(glmnet)
library(Metrics)
library(car)
library(MASS)
```

First task

Split the data set into training and test.

```
#Using the str function to present the values of the train and test data set.
str(train_x)
```

```
## num [1:545, 1:17] 1 1 1 1 1 1 1 1 1 1 ...
## - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:545] "Adrian College" "Agnes Scott College" "Albertson College" "Albertus Magnus College" ...
##   ..$ : chr [1:17] "PrivateYes" "Accept" "Enroll" "Top10perc" ...
```

```
str(test_x)
```

```
## num [1:232, 1:17] 1 1 1 1 1 1 1 1 0 0 ...
## - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:232] "Abilene Christian University" "Adelphi University" "Alaska Pacific University" "Albright College"
...
##   ..$ : chr [1:17] "PrivateYes" "Accept" "Enroll" "Top10perc" ...
```

The training and test data sets were generated in a 70/30 split for the number of applications using the College data set from the ISLR library, and the training and test data values were displayed using the str function. Our training data set comprises 545 rows, whereas our test data set only has 232 rows.

Second task

Estimate the lambda.min and lambda.1se values using the cv.glmnet method for ridge regression.

```
#So we want to validate that these above lines represent the minimum and 1 standard error, we can simple look at them here.
log(cv.ridge$lambda.min)
```

```
## [1] 5.937228
```

```
log(cv.ridge$lambda.1se)
```

```
## [1] 7.704869
```

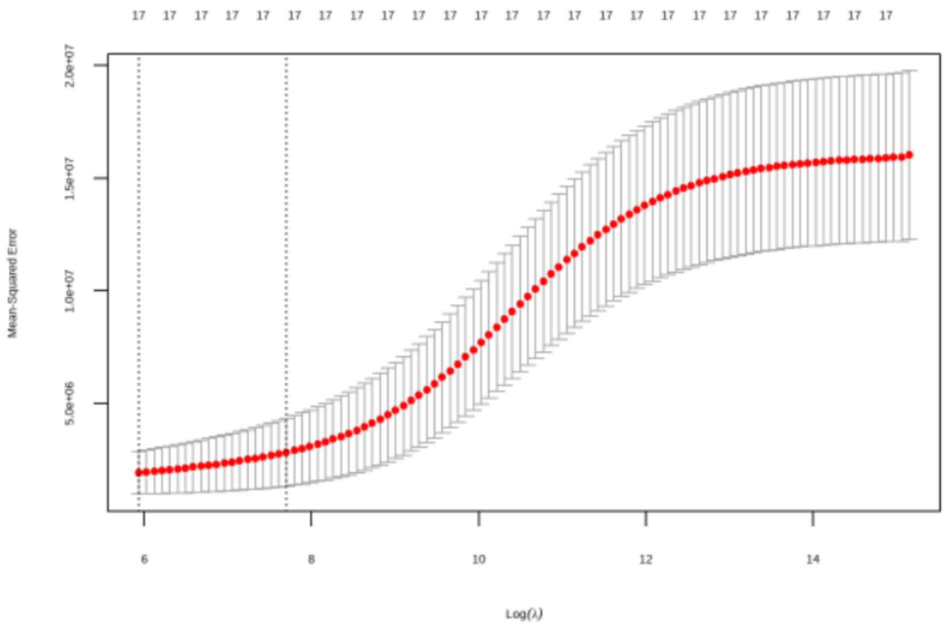
```
print(paste("The lambda.min =", round(log(cv.ridge$lambda.min), 5), " & the lambda.1st =", round(log(cv.ridge$lambda.1se),
5)))
```

```
## [1] "The lambda.min = 5.93723  & the lambda.1st = 7.70487"
```

Using the glmnet function from the glmnet library the lambda.min and lamda.1se values were obtained where we can notice that the lambda minimum value came out to be 5.93 while the lambda 1 standard error came out to be 7.704 which means that after 1 standard deviation from 5.93(minimum value) the value comes out to be 7.704(1 standard error).

Third task

Plot the cv.glmnet function results and offer an explanation for ridge regression.



The mean squared error is on the y axis, the log of lambda is on the x axis, and the non-zero coefficients are across the top (or the number of non-zero coefficients in the model for that particular value of lambda). These red dots now represent the error estimations. These two vertical dotted lines represent the plot's most important features. The minimal value of lambda is shown by the first line (left). As can be seen, the predictor model maintains roughly 17 variables. The second dotted line indicates lamda.1se, which is the largest/maximum value within one standard error of the lambda minimum. There are 17 non-zero coefficients in this model. As a result, just 17 variables remain. As a result, this is the simplest model (2nd) that performs almost as well as the best model (1st one).

Fourth task

Apply a Ridge regression model to the training data and report the results. Is there anything worth looking at?

```
## [1] "The regression coefficients for minimum value of lambda"
```

```
coef(model.minr)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -1.470628e+03
## PrivateYes  -5.312697e+02
## Accept       1.041548e+00
## Enroll       3.971297e-01
## Top10perc    2.586917e+01
## Top25perc    7.339721e-01
## F.Undergrad  5.418840e-02
## P.Undergrad  4.526920e-03
## Outstate    -2.930326e-02
## Room.Board   2.017959e-01
## Books        2.199339e-01
## Personal    -3.126986e-02
## PhD         -3.086166e+00
## Terminal    -3.630030e+00
## S.F.Ratio    9.141445e+00
## perc.alumni -8.047556e+00
## Expend       5.459078e-02
## Grad.Rate    1.288147e+01
```

```
## [1] "The regression coefficients for 1se value of lambda"
```

```
coef(model.1ser)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -2.322498e+03
## PrivateYes  -5.791444e+02
## Accept       5.498556e-01
## Enroll       7.509121e-01
## Top10perc    1.491471e+01
## Top25perc    7.644888e+00
## F.Undergrad  1.235120e-01
## P.Undergrad  9.112548e-02
## Outstate     8.419127e-03
## Room.Board   1.744033e-01
## Books        3.183808e-01
## Personal    -3.740664e-03
## PhD          3.150998e+00
## Terminal     2.685062e+00
## S.F.Ratio    1.044666e+01
## perc.alumni -9.357170e+00
## Expend       4.335692e-02
## Grad.Rate    1.288835e+01
```

We can utilize our values for the minimum and 1se to fit our models now that we have them, and we will use the glmnet function to do so. We set alpha = 0 for the ridge model (L2) and then search for regression coefficients for both the minimum and 1se models after fitting the model. The regression coefficients for both values are then presented using the coef function. We can see that the intercept and coefficient values for each variable column are different by looking at both tables.

Fifth task

Calculate the root mean square error to see how well the fit model performed against the training data (RMSE)?

RMSE Values

	Values
Ols/full model RMSE value	1,086.841
Training set RMSE value	1,574.648

Showing 1 to 2 of 2 entries

Previous 1 Next

We must first develop our ols model without regularization in order to obtain the RMSE result. Now we'll utilize this ols to calculate the RMSE number so we can see if our training and test data sets are overfit. Then, to compare with the ols model, the training data set RMSE is created. We can see that the value of our training is 1574.648.

Sixth task

Calculate the root mean square error to see how well the fit model performed versus the test set (RMSE)? Is your model over fitting?

RMSE Values

	Values
Ols/full model RMSE value	1,086.841
Test set RMSE value	1,331.766

Showing 1 to 2 of 2 entries

Previous 1 Next

Now we're calculating the RMSE value for the test data, which comes out to be 1331.766. The model is significantly overfitting the data since the RMSE value of the ridge predictions test set is lower than the training set. Overfitting occurs when the RMSE value difference is more than 0.2 or 0.5. When we compare it to the overall data set (ols), we can see that our overall is 1086 while our test is 1331, which is a significant difference, indicating that the data is overfit.

Seventh task

Estimate the lambda.min and lambda.1se values using the cv.glmnet method for lasso regression.

```
## [1] 3.309025
```

```
log(cv.lasso$lambda.1se)
```

```
## [1] 6.100037
```

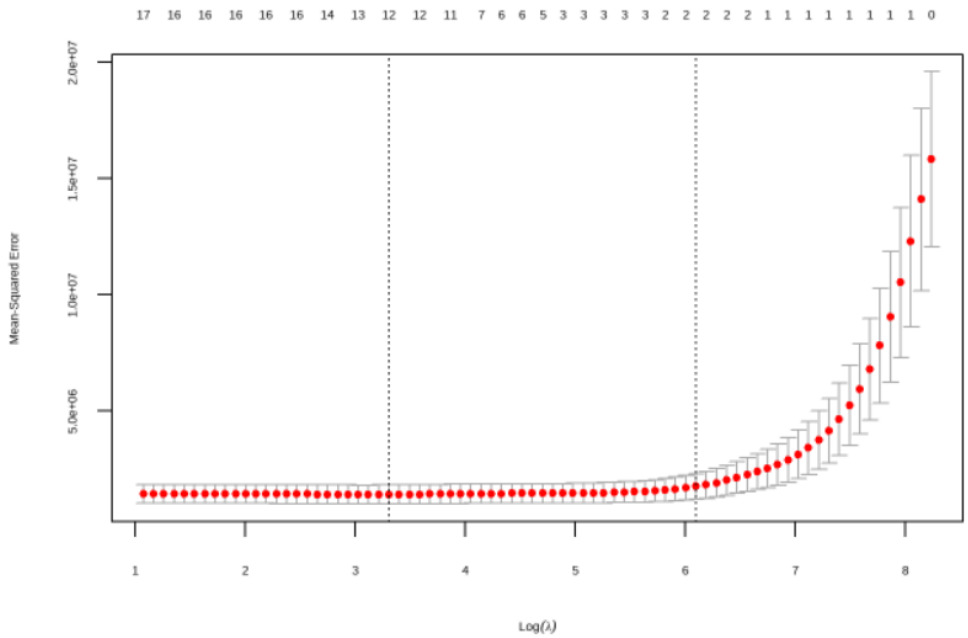
```
print(paste("The lambda.min =", round(log(cv.lasso$lambda.min), 5), " & the lambda.1st =", round(log(cv.lasso$lambda.1se),
5)))
```

```
## [1] "The lambda.min = 3.30903  & the lambda.1st = 6.10004"
```

Using the glmnet function from the glmnet library the lambda.min and lamda.1se values were obtained where we can notice that the lambda minimum value came out to be 3.30 while the lambda 1 standard error came out to be 6.10 which means that after 1 standard deviation from 3.30(minimum value) the value comes out to be 6.10(1 standard error).

Eighth task

Plot the cv.glmnet function results and offer an explanation for lasso regression.



The mean squared error is on the y axis, the log of lambda is on the x axis, and the non-zero coefficients are across the top (or the number of non-zero coefficients in the model for that particular value of lambda). These red dots now represent the error estimations. These two vertical dotted lines represent the plot's most important features. The minimal value of lambda is shown by the first line (left). As can be seen, the predictor model maintains roughly 12 variables. The second dotted line indicates lamda.1se, which is the largest/maximum value within one standard error of the lambda minimum. There are 2 non-zero coefficients in this model. As a result, just 2 variables remain and setting the coefficients for the rest of the variables as 0. As a result, this is the simplest model (2nd) that performs almost as well as the best model (1st one).

Ninth task

Apply a Lasso regression model to the training data and report the results. Is there anything worth looking at?

```
## [1] "The regression coefficients for minimum value of lambda"
```

```
coef(model.min)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -747.47632431
## PrivateYes  -369.39209902
## Accept         1.51095886
## Enroll        -0.38511301
## Top10perc     30.42800812
## Top25perc      .
## F.Undergrad    .
## P.Undergrad    .
## Outstate      -0.05598673
## Room.Board     0.11127013
## Books          0.06986633
## Personal       .
## PhD           -3.64612388
## Terminal      -1.35189923
## S.F.Ratio      .
## perc.alumni   -1.26437916
## Expend         0.04376610
## Grad.Rate      6.62615029
```

```
## [1] "The regression coefficients for 1se value of lambda"
```

```
coef(model.1se)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                    s0
## (Intercept) 18.064297
## PrivateYes   .
## Accept       1.283193
## Enroll       .
## Top10perc   13.534433
## Top25perc    .
## F.Undergrad  .
## P.Undergrad  .
## Outstate     .
## Room.Board   .
## Books        .
## Personal     .
## PhD          .
## Terminal     .
## S.F.Ratio    .
## perc.alumni  .
## Expend       .
## Grad.Rate    .
```

We can utilize our values for the minimum and 1se to fit our models now that we have them, and we will use the glmnet function to do so. We set alpha = 1 for the lasso model (L1) and then search for regression coefficients for both the minimum and 1se models after fitting the model. The regression coefficients for both values are then presented using the coef function. We can see that the minimum lambda coefficient has 12 variables with values and the remaining 5 are null or zero, whereas the 1 standard error coefficient has only 2 variables with values and the remaining 15 are zero, implying that the 1se performs as well as the minimum with only 2 coefficient variables.

Tenth task

Calculate the root mean square error to see how well the fit model performed against the training data (RMSE)?

RMSE Values

	Values
Ols/full model RMSE value	1,086.841
Training set RMSE value	1,259.233

Showing 1 to 2 of 2 entries   Previous 1 Next

We must first develop our ols model without regularization in order to obtain the RMSE result. Now we'll utilize this ols to calculate the RMSE number so we can see if our training and test data sets are overfit. Then, to compare with the ols model, the training data set RMSE is created. We can see that the value of our training is 1259.253.

Eleventh task

Calculate the root mean square error to see how well the fit model performed versus the test set (RMSE)? Is your model over fitting?

RMSE Values

	Values
Ols/full model RMSE value	1,086.841
Test set RMSE value	1,304.666

Showing 1 to 2 of 2 entries   Previous 1 Next

Now we're calculating the RMSE value for the test data, which comes out to be 1304.666. The model is significantly overfitting the data since the RMSE value of the lasso predictions test set is lower than the training set. Overfitting occurs when the RMSE value difference is more than 0.2 or 0.5. When we compare it to the overall data set (ols), we can see that our overall is 1086 while our test is 1304, which is a significant difference, indicating that the data is overfit.

Twelfth task

Which model performed better and why? Is that what you were hoping for?

The ridge model had a higher RMSE value than the lasso model, but when it came to overfitting from the original data, both models performed equally well. As a consequence, neither model was superior because they both produced overfitting findings. I was hoping that Lasso would do better, but that was not the case, since the test values for both models were nearly identical, with the exception of Ridge's enormous training value.

Thirteenth task

Fit a model after performing stepwise selection. Was this model more or less effective than Ridge regression or LASSO? Why do you choose one way over the other?

```
## [1] "The stepwise selection method = 1030.6232
```

Using the Stepwise selection, we got the value of 1030.62. When we compare this result to our original data value of 1086.841, we can see that it is closer than the ridge or lasso, but it still doesn't provide us enough information to say which approach is superior. The stepwise technique begins with no variables in the model and gradually adds them one by one while testing for improvement. It also checks for any previously introduced variables that are no longer needed in the model and eliminates them at each phase. The issues with stepwise model selection are more well understood and significantly worse than those with LASSO. LASSO is superior for selecting features or sparse models. Ridge regression, which utilizes all variables, may provide superior predictions.

CONCLUSION

In the end, I learned about glmnet, Metrics, and other libraries. Using the Ridge and Lasso model on the training and test data sets, worked with college data and performed regularization methods for models to characterize their link among variables and make effective predictions. I also learnt how to interpret the Ridge and Lasso plot to derive meaning from it. Finally, I studied about stepwise approaches and compared them to my model to see which was superior.

BIBLIOGRAPHY

1) S. (2021, April 27). Lasso Regression: Simple Definition. Statistics How To. https://www.statisticshowto.com/lassoregression/#:%7E:text=Lasso%20regression%20is%20a%20type,i.e.%20models%20with%20fewer%20parameters).

2) S. (2021a, February 6). *Ridge Regression: Simple Definition*. Statistics How To. https://www.statisticshowto.com/ridge-regression/