International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 11 Issue 1 ǁ January 2023 ǁ PP. 360-367

A Review of Techniques for Regularization

[1]Jude Chukwura Obi and [2]Ibebuike Chinenye Jecinta
[1,2] Department of Statistics,
Chukwuemeka Odumegu Ojukwu University, Anambra State, Nigeria
Corresponding Author: Jude Chukwura Obi
Email: oj.obi@coou.edu.ng

ABSTRACT: *Regularization is a technique used for solving regression problems in the presence of multicollinearity which ultimately leaves behind a singular design matrix. Recall that once the singular design matrix is available, the ordinary least squares (OLS) cannot produce any solution for a regression problem. This study, therefore reviewed three different techniques for regularization namely the ridge (L2 penalty), lasso (L1 penalty) and elastic net. We emphasized on how the threes techniques use lambda and alpha as their shrinkage or turning parameter that decides how we can penalize the OLS cost function. The L1 and L2 penalty regularizes in a similar but different way.Their regression coefficient differs because L2 penalty add lambda and the sum of square of coefficients, and L1 add lambda and the sum of absolute value of coefficients as their penalty terms, while elastic net is the combination of both L1 and L2. We however, carried out our analysis on 14 different datasets using R software as our statistical package to analyse these datasets.Using cross validation approach, we select the optimal value of lambda and alpha for elastic net. We then obtained R-squared*(R^2), *mean squared error* (MSE), *and root mean squared error (RMSE) of these techniques as the output, and observed that, the output of these techniques on the studied dataset are nearly the same remarkably. Using permutational multivariate analysis of variance (because MANOVA failed the shapiro-wilks test of normality, indicating that the assumptions are not met) to test the hypotheses of null and alternative, to decide if there are significant difference in their performances, for L1, L2 and elastic net. The result presented a p-value of 0.868, which shows that we cannot reject the null hypothesis, and there exist no significant difference between these techniques. The result of this analysis further solidifies that both ridge, lasso and elastic net regression tally in their output predictions. We however, carried out a simulation study to demonstrate that OLS does not perform when there are problems of multiple correlation, and singular design matrix (when* $p \cong n,$ *and when* $p \gg n$). *Via the use of regularizationtechniques, these penalized OLS estimators (lasso, ridge regression and elastic net) were able to take care of the deficiency of OLS estimators when multicollinearity is available.*

KEYWORDS*: Mulitcollinearity, Ordinary Least Square, Regularization, Ridge regression, Lasso regression, elastic net regression.*

Date of Submission: 11-01-2023 Date of acceptance: 27-01-2023

I. INTRODUCTION

The term multiple linear regression is a statistical technique that uses multiple input variables to predict the output or response variable. The multiple linear regression has the model

$$Y = \beta_0 + \beta_1 X_1 + \cdots \beta_n X_n + \varepsilon, \quad (1.1)$$

where

Y = target variable or response variable. β_0 = intercept term or the value of y when other independent values are zero. β_1= coefficient of regression for the X_1 variable. β_n= coefficient of regression for the n^{th} variable.X_1 = independent, input, predictor or explanatory variable. ε = error term that associates with the model.

In other to reduce possible biasness, numerous number of predictor variables may be introduced in the model, which may lead to a serious concern of multicollinearity (i.e. the explanatory variables are highly correlated, with each other) among the predictor variables. This poses a problem because if the data shows multicollinearity, the problem of figuring out the specific variable that contribute to the variance in the dependent variable arises.Also, in a multiple regression analysis it is always the case that **p** << **n** (where **p** is the number of predictors, and **n** is the number of observation), because, the model parameters tend to have low variance and will hence perform well on test data. But when **p** ≅ **n** or **p**>>**n** it becomes a problem, when

- **p** ≅ **n**: This means that, the variability of the least square coefficient of estimation is much resulting in overfitting and poor prediction of observation not used in model training.

- **p>>n**: This mean that, number of predictors p, is larger than the number of observations of the sample, *n*.).

In this case, the OLS (ordinary least square regression) becomes unworkable and $X^T X$is singular and its inverse ceases to exist, meaning that we are stuck and can't move further. Penalized regression techniques comes into play with it parameters to deal with cases like this because, penalized regression methods do not clearly select the variables; instead they minimize the Regression Sum of Square by using a penalty on the size of the regression coefficients.

1.1 REGULARIZATION

Regularizationis a shrinkage regression procedure that is implemented when it not possible to use OLS because of the problem of singular design matrix. It makes little adjustment to the model such that, it performs well on both training and test data.

1.2.1 How Does Regularization Work?

The loss function in (1.2) for the estimated parameters is known as RSS (residual sum of square). This model parameters are adjusted by trading some bias at the expense of variance so as to reduce this loss function. The RSS is given as:

$$\text{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \qquad (1.2)$$

Where n = number of observation and p= number of predictors.
Regularization here will shrink or adjust the coefficients in (1.2) based on the training data.

1.3 Bias-Variance Trade-Off in OLS and Regularization

The concept of bias and variance emanated from (Geman et al. 1992), which indicates that there is a trade-off required between the ability of a model to reduce bias (maximise accuracy) and reduce variance (maximise precision), when there is an attempt to infer result from a stochastic process.The bias of a model is defined as a measure of how far or close the estimated values is to the actual values. It measures the accuracy of our estimates or model, and describes how well our model fits the training dataset.It can as well be defined as the difference between the expected estimate and the true population parameter we are estimating.

Variance is a measure of variation in our prediction. It measures the spread, or uncertainty, in our estimates. It can also be defined as the measure of difference in fit between the training dataset and the test dataset.

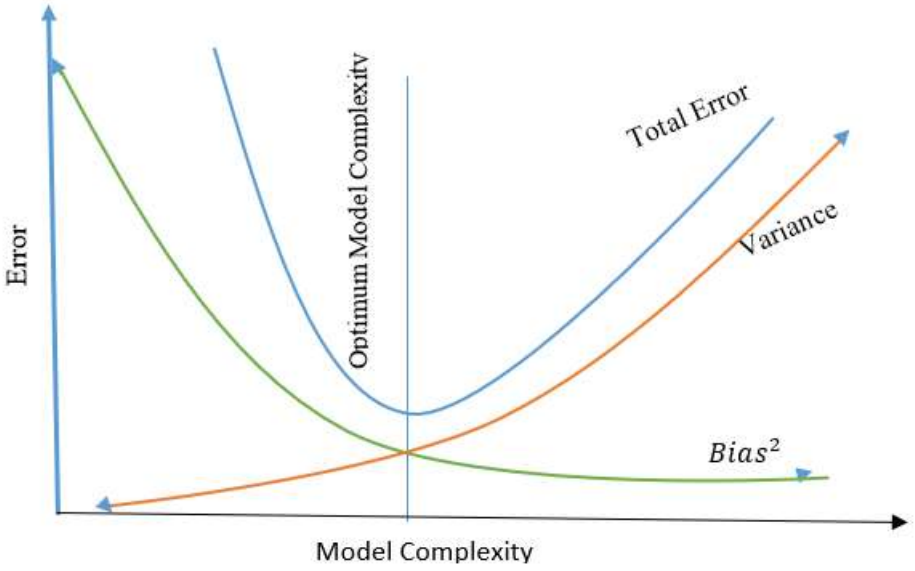

Figure 1.1: Graphical illustration of the input of bias and variance to general error (Scott Fortmann-Roe, 2012). On the right-hand side of the graph, lies the unbiased estimates and large variance estimate, and on the left is the bias and low variance estimates which are both far from optimal.

Figure 1.1 illustrates bias variance trade-off , where increasing model complexity has the eff ect of reducing bias in the model while increasing variance at the same time (Scott Fortmann Roe, 2012).

As the number of predictors increases, the bias decreases but the variance estimates increases and vice visa. This shows that large values result in poor prediction, and fewer values result in poor accuracy of the model. As we can see, there is an optimal point in the graph in which our total error is minimised between some intermediate model complexity where the balance between bias and variance is obtained and relatively low. However, if our model complexity goes beyond it, it leads to overfitting, and if it falls short of it, leads to under fitting. Regularization comes into play because it able to improve the model by trading some variance at the cost of introducing some bias. This is achieved by introducing additional terms that penalizes or shrinks those coefficients that do not explain a greater number of the variance in the model.

1.4 Regularization Techniques in Machine Learning

Three regularization techniques are going to be reviewed here and they include the:

- Ridge regression
- Lasso regression and
- Elastic net regression

1.4.1 Ridge Regression

A regression techniques that uses L2 penalty is called Ridge Regression (Hoerl and Kennard, 1970). Supposing n is the sample size and p is the number of predictor variables in a dataset. The OLS estimator $\hat{\beta}$ is defined only if the design matrix $X^T X$ is invertible, otherwise the design matrix is singular. Ridge regression can be a useful instrument when they have to deal with multicollinearity. The typical OLS function is constrained by the L2-penalization, which shrinks coefficients towards zero. The illustration on ridge regression is shown in (1.4)

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_0 x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta^2 \qquad (1.4)$$

Equation (1.4), modifies the sum of square residual by introducing a shrinkage parameter. Here, the regression coefficients are estimated by reducing this function. Lambda being the tuning parameter or hyper parameter decides the extent we penalize the workability of our model. As coefficients increases, our model complexity increases. However, to minimize the above function, these coefficients will be reduced as this penalty terms are added. Therefore, this shrinkage procedure is adopted in other to shrink or penalize higher coefficients.

1.4.2 The Lasso (Least Absolute Shrinkage and Selection Operator)

A shrinkage regression procedure that uses L1 penalty is known as Lasso. The lasso is an acronym for "least absolute shrinkage and selection operator" (Tibshirani, 1996). This acronym comes from its functionality as a shrinkage procedure and a variable selection technique.The L1 penalty forces some of the coefficient estimates of regression to be equally zero, this happens when the tuning parameter λ is large. The L1 penalty function of the lasso is shown in (1.5).

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_0 x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|. \qquad (1.5)$$

Equation (1.5) shows how the function is minimized using lasso techniques. This is also known as the L1 normalization (norm). It's clear that, there are variation between ridge and lasso regression in terms penalizing coefficients that are high. Lasso uses lambda and the modulus of $\beta_j$instead of squares ofβ_j, as its penalty.

1.4.3 Elastic Net Regression

The elastic net represents a modification to ridge and lasso(Zou and Hastie, 2005). It is able to provide a spare solution, and eliminates the problem that occurs when lasso is used on data sets with highly correlated variables. Elastic net uses the penalties from both ridge and lasso techniques to regularize regression model. The technique combines both lasso and ridge regression by learning from their weaknesses. Elastic Net aims at minimizing the following loss function:

$$L_{enet}\left(\hat{\beta}\right) = \frac{\sum_{i=1}^{n}\left(y_i - x_i^j\hat{\beta}\right)^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j=1}^{m}|\hat{\beta}_j|\right), \qquad (1.6)$$

where α is the turning parameter between ridge ($\alpha = 0$) and lasso ($\alpha = 1$). Thus, there are two turning parameters λ and α.

1.5 Selection of the Turning Parameter

Unlike the OLS, the ridge and lasso estimator relies on a turning parameter or hyper parameter λ, used to regulate the balance between bias and variance. As the λ value increases, the value of coefficients decrease. Thisincrease in λ is important because it reduces the variance and as well avoid overfitting, without losing properties beneficial to the model.If lambda value is equated zero, it takes us back OLS. Moreover, if the value of lambda is very large it leads to under-fitting thereby adding too much weight to the cost function. Having said this, the choice of lambda value is paramount. This techniques works very well and help to avoid over-fitting problem.

1.6 Cross validation

This is a procedure used in other to obtain optimum value of lambda. That is, to find λ with the lowest mean squared error (MSE), by repeatedly holding out a subset of the observations, and applying the chosen method to predict the held-out outcome.There are many cross validation techniques but our focus is on the k-fold CV.The K-fold cross-validation procedure partitions the data set roughly into K different subsets, for training and testing.Each partition is term a "fold" of the dataset, typically five or ten folds are practically indicated as good for computational burden. The K-1 folds is the training set and the kth fold is used for test set (Clarke, Fokoue, & H. H. Zhang, 2009). This process is repeated k times, such that each fold is used for testing exactly once using each of the K subsets as a validation set, thus yielding K estimates of the MSE for each parameter value. The K-fold estimate (1.8) is simply the average value of the K estimates.

$$\mathrm{KCV} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i} - \hat{\mathrm{y}})^2, \tag{1.8}$$

where $\hat{y}$ is the predicted or fitted values, and y_i is the observed values on the k^{th} fold and i is the number of rounds in model training and validation. The optimal $\hat{\lambda}$ is chosen as:

$$\hat{\lambda} = \arg\ \min \mathrm{KCV} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\mathrm{y_i} - \hat{\mathrm{y}}^{-\mathrm{k(i)}}\right)^2. \tag{1.9}$$

The model with the lowest MSE, becomes the CV prediction error and it λ becomes the optimal value.

II. AIM AND OBJECTIVES

2.1 Aim

To investigate different ways of achieving regularization, with particular reference to a situation where we have singular design matrix.

2.2 OBJECTIVES

- To develop a model that generalizes well on training and test set.
- To know if the techniques differ from their performances or tally in output prediction.
- To know whether the regularized regression leads to better prediction in general.
- To examine the effect, if any, of the different regularization techniques discussed so far on the data (to know how they regularize).

III. RESEARCH METHODOLOGY

The following are material and methods used:

- R software, glmnetfunction(), cross validation (k-fold) approach.
- Ridge regression, lasso regression, elastic net regression.
- R-squared, mean squared error, root mean squared error.
- shapiro-wilks test of normality, nonparametric MANOVA(adonis2).

IV. DATA ANALYSIS ON RIDGE, LASSO AND ELASTIC NET REGRESSION

A total of 14 different real-world datasets and three simulated datasets will be used for this study. The real-world datasets were accessed from the internet and they includeMtcars dataset(Henderson and Velleman, 1981), R built in data, Abalone dataset, Cancer data, Diamond data set, Kc house dataset, Bodyfat dataset, Real estate valuation, Concrete dataset, Forestfire dataset, Heart failure dataset, Heartdisease dataset, Steel industry dataset, Garment dataset and Fish dataset.These datasets can be accessed at
https://archive.ics.uci.edu/ml/datasets/abalone;https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
andhttps://www.kaggle.com/datasets/shivam2503/diamonds?resource=download

4.1 DATA ANALYSIS RESULTS

		Ridge Regression			Lasso Regression			Elastic Net		
Dataset	Dataset Dim	R^2	MSE	RMSE	R^2	MSE	RMSE	R^2	MSE	RMSE
Mtcars	32×11	0.97	453.2	21.29	1.0	14.10	3.76	1.0	14.38	3.79
Abalone	4177×8	0.51	5.43	2.33	0.49	5.31	2.30	0.49	5.31	2.31
Cancer	3047×28	0.48	392.24	19.80	0.49	401.90	20.05	0.53	375.26	19.37
Diamonds	53940×7	0.83	2742750	1656.13	0.86	2251514	1500.5	0.86	2243514	1497.85
KC House	21613×17	0.66	4.6E+10	214768	0.66	4.6E+10	213638	0.66	4.6E+10	213638
Bodyfat	252×15	0.96	3.24	1.80	0.99	0.60	0.77	0.99	0.60	0.77
Real Estate	4147	0.98	4.11	2.03	1.0	0.19	0.44	1.0	0.20	0.44
Concrete	1030×9	0.63	106.479	10.31	0.63	106.807	10.33473	0.57	121.62	11.03
Forest fire	517×11	1.0	34.24	8.15	1.0	3.53	1.88	1.0	3.60	1.90
Heart	1025×14	0.53	0.12	0.35	0.58	0.12	0.35	0.47	0.14	0.37
Heart Failure	299×13	0.36	0.17	0.41	0.38	0.16	0.40	0.38	0.16	0.40
Steel Industry	35040×5	0.91	105.60	10.28	0.91	100.75	10.04	0.92	100.75	10.04
Garment	1197×10	0.28	0.02	0.15	0.28	0.02	0.15	0.29	0.02	0.15
Fish	159×7	0.87	19522.5	139.72	0.88	18924	137.56	0.88	18924.8	137.57

Table 4.1: Recording on the result of $R^2, MSE, RMSE$ values for ridge, lasso and elastic net.

Table 4.1 represents the result of the R squared, mean squared error and root mean squared error on analysis of ridge, lasso and elastic net regression using from R software, on 14 different dataset. The result indicates that the R^2output of the three techniques on all datasets is in conformity with each other or nearly the same, but the MSE and RMSE produced close result for each techniques except for mtcar dataset where the mean square error output, of ridge regression by 420.82 difference larger than the others. Now the situation is to ascertainif these techniques differ from their performances based on their output. In other to determine this, we will use multivariate analysis of variance to test the null hypothesis of equal mean across the techniques and the alternative hypothesis which says otherwise. Considering the P-value result obtained from the normality test (see Appendix A.5) using Shapiro-Wilk test, the normality assumptions of MANOVA are not met. However, using permutational MANOVA,the test procedure goes as follows:

$$\mathrm{H}_0: \begin{pmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{13} \end{pmatrix} = \begin{pmatrix} \mu_{21} \\ \mu_{22} \\ \mu_{23} \end{pmatrix} = \begin{pmatrix} \mu_{31} \\ \mu_{32} \\ \mu_{33} \end{pmatrix},$$

$$\mathrm{H}_1: \text{At least one } \boldsymbol{\mu}_{\mathrm{i}} \neq \boldsymbol{\mu}_{\mathrm{i}'}\ ;\ \mathrm{i}, \mathrm{i}' = 1,2,3.$$

Here, the null hypothesis states that there is no significant difference between the three methods, while the alternative hypothesis argues otherwise.Using PERMANOVA test, The P-value of 0.868 wasobtained, which indicates that there is no reason to reject the null hypothesis.

4.2 A Simulation Study

The purpose of this simulation study is to show that OLS offers no solution to a regression problem when collinearity is present. It is often the case when $p \cong n$, or $p \gg n$. The dataset used here is simulated using R statistical software. Three models are presented as follows:

(a) The first simulation shows that the OLS model produces no result when p ≅ n. The stimulated dataset is with dimension 30×30,meaning thatn = 30 and p =30.

Model summary: Residual standard error: NaN on 0 degrees of freedom. Multiple R-squared: 1. Adjusted R-squared: NaN. F-statistic: NaN on 29 and 0 DF, p-value: NA

(b) The second simulation shows similar result when p ≫ n. The stimulated dataset is with dimension 30×40, where n = 30 and p =40.

Model summary: Residual standard error: NaN on 0 degrees of freedom. Multiple R-squared: 1, Adjusted R-squared: NaN. F-statistic: NaN on 29 and 0 DF, p-value: NA

(c) The third simulation shows similar results as in (a) and (b) above. The stimulated dataset has dimension $8 \times 50$where n = 8 and p =50.
Model summary: Residual standard error: 3.885e-14 on 42 degrees of freedom,Multiple R-squared: 1, Adjusted R-squared: 1, F-statistic: 2.121e+28 on 7 & 42 DF, p-value: < 2.2e-16.

Based on the p-value, R-squared and adjusted R-squared of the model summaries of the three simulated datasets, the OLS model here has no solution for a regression problem when datasets are highly correlated. However, it has been equally shown that the use of regularization techniques is able to correct this core weakness of the OLS, given highly correlated datasets. Appendix B consists of the R codes used for datasets simulations.

V. SUMMARY/CONCLUSION

So far, a total of 14 processed real-world datasets were analysed using R statistical software. Cross validation was used with glmnetfunction() contained in R software, to automatically select the optimum value of lambda and alpha in elastic net.With alpha as a turning parameter, to ascertain which techniquewe are concerned with, $\alpha = 0$if we are concerned with ridge and for lasso, $\alpha = 1$. In the case of elastic net, alpha is between 0 and 1. We proceeded with the analysis, and obtained R^2, MSE, and RMSEfor each regularization technique, on every dataset used in the study. The second phase is the use of MANOVA to run a test of hypothesis for decision making on whether to accept or reject the null hypothesis. At a p-value of 0.9998, the null hypothesis could not be rejected which goes further to support the claim that the three different techniques do not vary in the output generated. However, on checking the normality test assumption for the MANOVA test, the null hypothesis was rejected meaning that the datasets for the study do not comply with the normality assumption. Furthermore, using a nonparametric permutational MANOVA test, we obtained a p-value of 0.868, meaning that we still cannot reject the null hypothesis.For this reason, it follows that any of the three regularization techniques is equally as good as the other. Hence, we conclude that with respect to the given datasets for this study, the output of ridge, lasso and elastic net regression do not differ from one another.

REFERENCES

[1]. Clarke, B., Fokoue, E., & Zhang, H. H. (2009). Principles and theory for data mining and machine learning (pp. 304-310). New York: Springer.
[2]. Fortmann-Roe, S. (2012). Understanding the Bias-Variance Trade-off. URL: http://scott.fortmann-roe.com/docs/BiasVariance.html(hamtad2019-03-27).
[3]. Friedman, J., T. Hastie, and R. Tibshirani (2009). glmnet: Lasso and elastic-net regularized generalized linear models. R package version 1(4).
[4]. Geman, S., Beienenstock, E., and Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. Neural Comput. 4, 1-58. Doi:10.1162/neco.1992.4.1.1
[5]. Hastie, T., &Tibshirani, R. (2001). Friedman. The Elements of Statistical Learning," Springer, 52.
[6]. Henderson, H. and Velleman, P. F. (1981). Building multiple regression model interactively. Biometrics, **37**, 391-411.
[7]. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.
[8]. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2), 301-320.

APPENDIX A

A.0 Codes and Results of Data Analysis

Real Estate Valuation Dataset

```
>rm(list = ls()).  >dat = read.csv(file.choose(), header = T); dat[1:2, ].  >> #creating training and test set
> dd = sort(sample(nrow(dat), nrow(dat)*0.7)).  >>Xtrain = dat[dd,]  >>Xtest = dat[-dd,]
> ## Forming data matrix.  >> X = data.matrix(Xtrain[, -1])  >> y = Xtrain$Y.house.price.of.unit.area
>X.new = data.matrix(Xtest[, -1])  >>y.new = Xtest$Y.house.price.of.unit.area
> ## Parameters Definition  >> n = nrow(X.new)  >> p = ncol(X.new)
```

A.1 ## Ridge Regression

```
> library(glmnet).  >>set.seed(123).  >>cv_model = cv.glmnet(X, y, alpha = 0, family = "gaussian")
>best_lambda = (cv_model$lambda.min); best_lambda
> #find coefficients of best model
>best_model = glmnet(X, y, alpha = 0, type.measure="mse", lambda = best_lambda); best_model>coef(best_model)
> #use fitted best model to make predictions
>y_predicted = predict(best_model, s = best_lambda, newx = X.new)
> #find SST and SSE.  >> #find R-Squared, mse, rms.  >>sst = sum((y.new - mean(y.new))^2); sst
>sse = sum((y_predicted - y.new)^2); sse.  >>rsq=(1 - sse/sst); rsq
```

>mse = sum((y_predicted - y.new)^2)/(n-p-1); mse. >>rmse = (sqrt(mse)); rms
A.2 ## Lasso Regression (use alpha = 1, set.seed(123) and apply same procedure as ridge) eg
>cv_model = cv.glmnet(X, y, alpha = 1, family = "gaussian"). >> #find SST, SSE, R-Squared, mse, and rms
A.3 ##Elastic Net Regression. >> library(caret). >>set.seed(123)
> Control = trainControl(method = "cv", number = 10)
>elastic_model = train(Y.house.price.of.unit.area ~., data = dat, method = "glmnet",
+ trControl = Control, tuneLength = 10)
> # Best tuning parameter. >>elastic_model$bestTune
>coef(elastic_model$finalModel, elastic_model$bestTune$lambda)
> #use fitted best model to make predictions
>y_predicted = predict(best_model, s = best_lambda, newx = X.new). >> #find SST, SSE, R-squared, mse, rms
A.5 Shapiro-Wilk normality test
>shapiro.test(dd$**RS**); data: dd$RS. >>W = 0.88084, p-value = 0.0004035
>shapiro.test(dd$**MSE**); data: dd$MSE. >>W = 0.28443, p-value = 4.879e-13
>shapiro.test(dd$**RMSE**); data: dd$RMS. >>W = 0.28739, p-value = 5.199e-13
A.6 Performance of Nonparametric ANOVA
> dd = read.table(file = "clipboard", header = T); dd[1:6,]
Group RS MSE RMS
1 Ridge 0.970 4.532000e+02 21.29
> library(vegan). >> ## Create the Y matrix of variables under comparison:
> Y = dd[, c("RS", "MSE", "RMS")]; head(Y)
A.7 Nonparametric Test
> ## Perform a one-way PERMANOVA: >> ## See https://f-santos.gitlab.io/2020-05-07-npmanova.htm
observe we use adonis2 and not adonis as contained in the website
> adonis2(Y ~ dd$Group, method = "euclidean", permutations = 999)
Permutation test for adonis under reduced model. Terms added sequentially (first to last)
Permutation: free. Number of permutations: 999
adonis2(formula = Y ~ dd$Group, permutations = 999, method = "euclidean")
DfSumOfSqs R2 FPr(>F)
dd$Group 2 1.1180e+16 0 **0 0.868**
Residual 39 5.8442e+21 1 Total 41 5.8442e+21 1

APPENDIX B
B.0 SIMULATED DATA ANALYSIS SOLUTIONS AND RESULTS
B.1.0 Simulated Dataset with Dimension 30x30
rm(list = ls()). >>## Data dimension 30X30. >>library(MASS)
>Mean.vector = sample(1:100, 30, replace = F). >> library(clusterGeneration)
> Sigma = genPositiveDefMat(dim = 30, covMethod = "unifcorrmat")[[2]]
> n = 30. >> # create multivariate normal distribution
>Multi.sample = round(mvrnorm(n, mu = Mean.vector, Sigma = Sigma), digits = 2)
>dat = data.frame(Multi.sample). >>dat$y = round(abs(dat$X1+2-3^3*sqrt(5)), digits = 2); dat[1:2,]
> mod = lm(y ~., data = dat); summary(mod)
Call: lm(formula = y ~ ., data = dat). >>Residuals: ALL 30 residuals are 0: no residual degrees of freedom!
Coefficients: (1 not defined because of singularities)
Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.837e+01 NA NANA
X1 1.000e+00 NA NANA
Residual standard error: NaN on 0 degrees of freedom. Multiple R-squared: 1, Adjusted R-squared: NaN
F-statistic: NaN on 29 and 0 DF, p-value: NA

USING RIDGE, LASSO AND ELASTIC NET REGRESSION
B.1.1 ##Fit Ridge Regression Model
> library(glmnet). >>set.seed(222). >> Y = dat$y. >> X = as.matrix(dat[, 1:30])
> model = glmnet(X, Y, alpha = 0). >>cv_model = cv.glmnet(X, Y, alpha = 0)
>best_lambda = (cv_model$lambda.min); best_lambda
>best_model = glmnet(X, Y, alpha = 0, lambda = best_lambda). >>coef(best_model)
B.1.2 ##fit Lasso Regression (use alpha = 1, set.seed(222)and repeat the process)
B.1.3## Elastic Net Regression (use library(caret), set.seed(222) and apply same procedure) and

```
> #Obtain coef(elastic_model$finalModel, elastic_model$bestTune$lambda)
```

B.2.0 Second Simulated Dataset on 30X40 Data dimension

```
>rm(list = ls())  >>## Data dimension 30X40.    >>library(MASS)
>Mean.vector = sample(1:100, 40, replace = F).       >> library(clusterGeneration)
> Sigma = genPositiveDefMat(dim = 40, covMethod = "unifcorrmat")[[2]].   >> n = 40
> # create multivariate normal distribution
>Multi.sample = round(mvrnorm(n, mu = Mean.vector, Sigma = Sigma), digits = 2)
>dat = data.frame(Multi.sample).      >>dat$y = round(abs(dat$X1+2-3^3*sqrt(5)), digits = 2); dat[1:2, ]
>dat = dat[1:30, ]; dim(dat).       >>[1] 30 41.    >> mod = lm(y ~., data = dat); summary(mod)
```