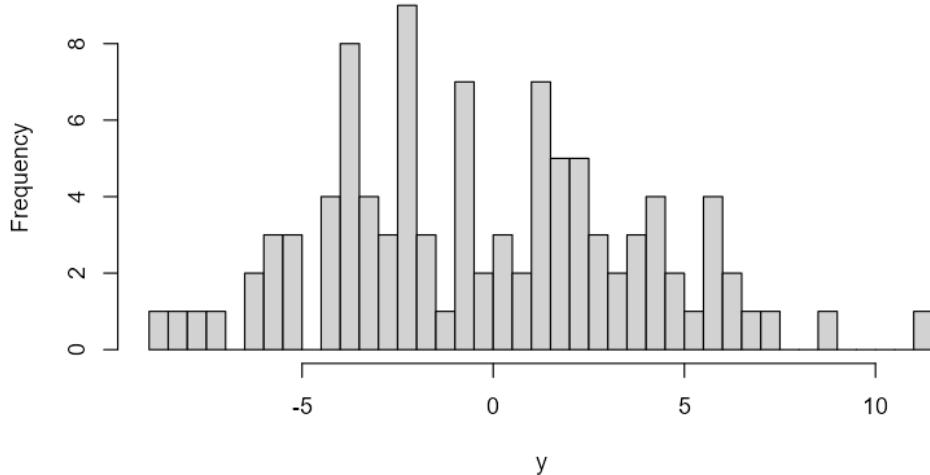


Q1A::

Firstly, we perform EDA to find out what type of data we're working with. "df" is 100 rows and 100 predictor columns, not including the target variable y.

We create a histogram & observe that y is approximately normal, then do an 80-20 test split.

Next, I standardized the predictor variables around the mean and standard deviation of each X1:X100 in the training set, then the test set using the same parameters as the training set.

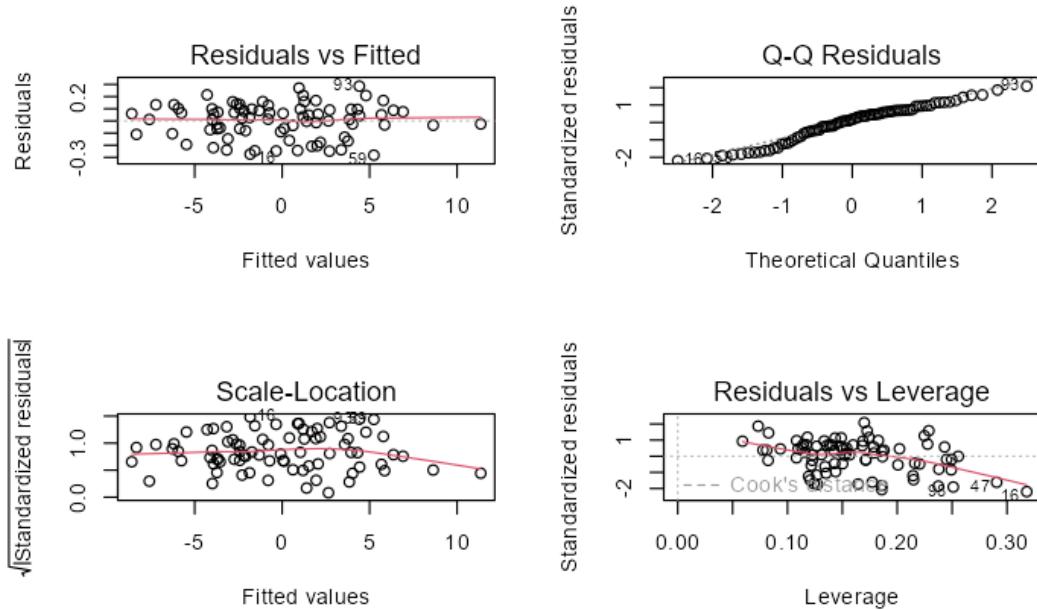


Next, we fit the full model. Due to the large number of predictors, $p > n$ and the model is undefined- We will need to find a new fitting method. Stepwise selection yields an undefined AIC, so it's not a mathematically valid method of fitting a better model.

We need to move on to lasso/ridge regression in order to select the best parameters. First I tried lasso regression and retrieved lambda min = 0.03025307. The fitted model with the lasso parameters [X1, X14, X15, X16, X20, X26, X29, X51, X58, X62, X64, X90]. Next, I tried Elastic

Net with lambda min = 0.06012679. Elastic net yielded 23 parameters, and its MSE was nearly double that of lasso. We also fit a ridge regression.

We fitted our models with the selected parameters, then we created our residual and Q-Q plots for lasso since it had the best results- they show even spreads on regular and standardized residuals. The Q-Q plot verifies approximate normality of residuals with some slight bending.



We plot predicted vs. actual y values in both the test and training set [Fig. 1, 2]. We created an L-R plot to look for any missed outliers and high leverage points [Fig. 3], however there are no anomalous values present.

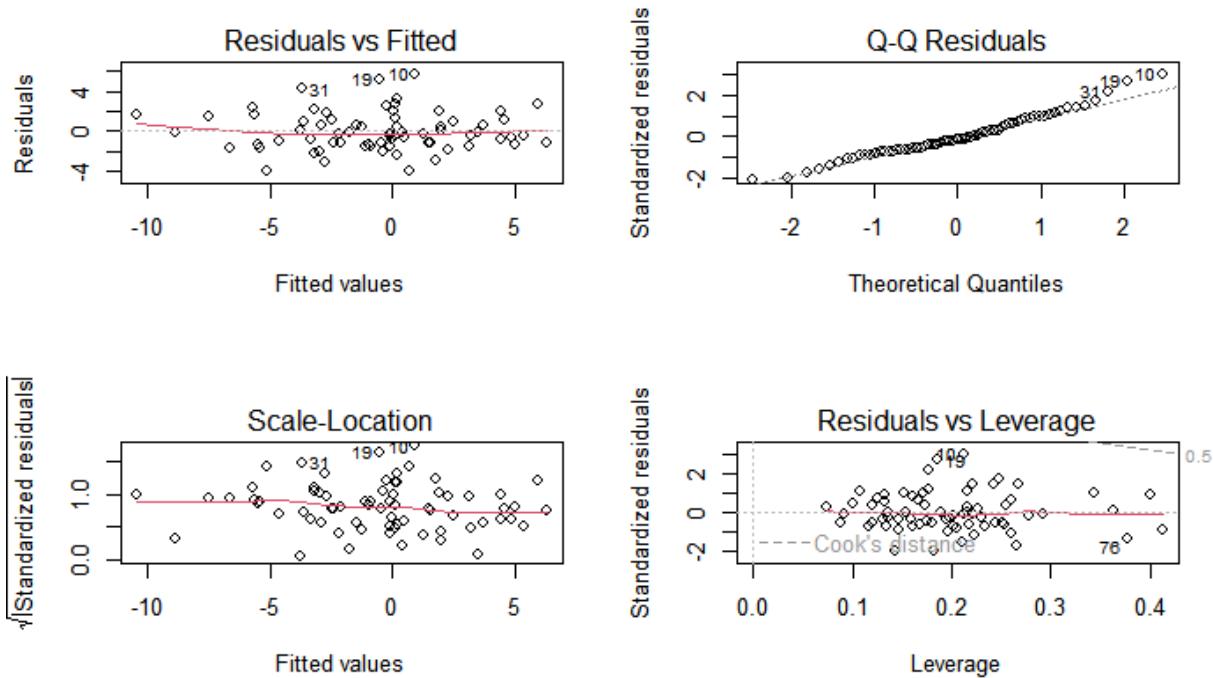
Lastly, we tested for multicollinearity. First, we tested VIFs: X1 and X51 are high and condition number = $16.4 > 15$, indicating presence of some multicollinearity. We tried the same model while dropping predictors X1 and X51. The resulting R^2 changes were poor. When we remove only X51, we have a model with low collinearity (VIFs are about 1) and a strong R^2 . The following are the test set results:

	MAE	R²	MSE	RMSE
Lasso (Reduced)	0.26	0.9930	0.10	0.33
Enet	4.42	-0.7204	26.78	5.17
Ridge	3.54	0.0079	15.43	3.92

In conclusion, the reduced lasso (remove X51) model fits very well. 99.3% of the variation in y is explained by our model. RMSE = 0.3295 means that our residual error is only about 0.33 on average while the MSE and MAE are both comparatively low. Since our reduced lasso model accounts for descriptive statistics, outliers/high influence points, multicollinearity, and accuracy/overfitting, we can safely conclude this is the best model.

Q1B::

For Q1B, we repeat the same process & model evaluation. In 1B, we have a very similar dataset to 1A. We have 100 rows, 100 predictor columns, and 1 target continuous variable, y. We started by producing a histogram and looking at the distribution of y, it seems approximately normal. Since we have a small dataset and an 80% split yielded poor results, we will use a 70-30 split. We used the same standardization as in 1A. We run the same lasso, ridge, and elastic net regression & fit the corresponding model parameters.



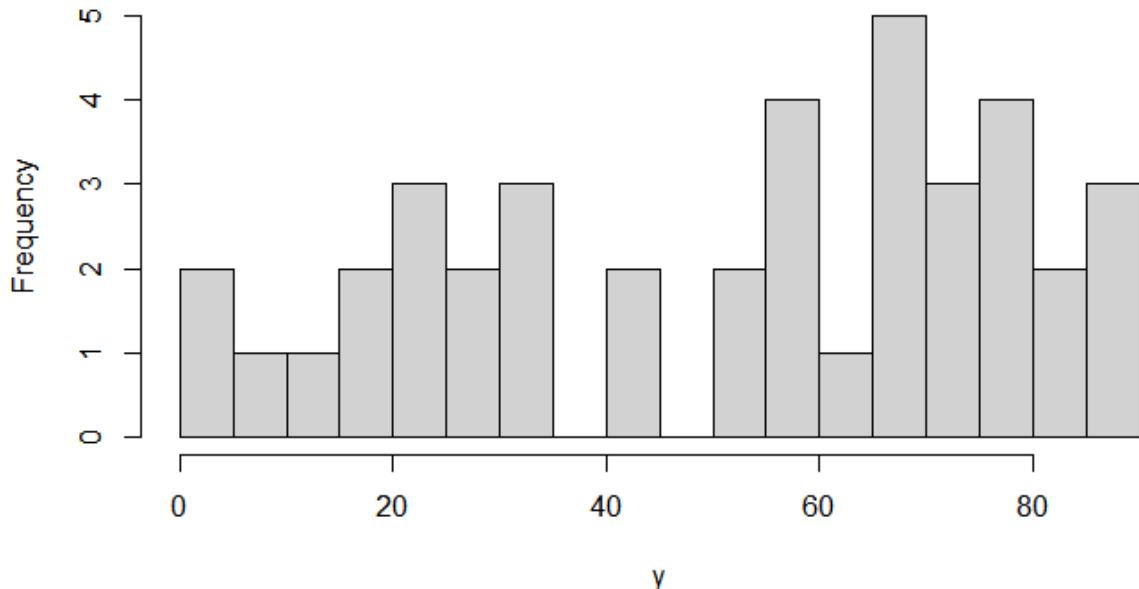
Our lasso yielded the best results again & our diagnostic plots reveal normality with some flaring at the tails of our Q-Q residual plot. Our residual lots confirm homoskedasticity. We tested for multicollinearity and there isn't evidence of any strong dependencies between predictors.

	MAE	R ²	MSE	RMSE
Lasso	2.15	0.664	7.98	2.82
Enet	4.265	-0.147	27.24	5.21
Ridge	4.63	-0.388	32.95	5.74

Our above test set results reveal lasso as the best model. While the fit is not as good as in 1A, 66.4% of variation is explained by our model, which could be very good in context.

Q2::

In question 2, we have 40 rows of data with 4 predictor columns (Lat, LBW, ELE, SMR) and the target variable, lfc. “Lfc” stands for “lightning flash count,” an integer count variable between 1 and 89. From our histogram, we can observe a somewhat uniform distribution of lfc, slightly skewed left.



Since we have a count variable, we will start by using the Poisson loglinear model. First we fit the full model and then the null model. We run a likelihood ratio test and compare AICs between the two models. Our p-value << 0.05 and our AIC decreased from 886.1159 to 826.4404. The full model is definitely an improvement by these metrics, however we calculated our dispersion parameter and got 17.02557- indicating significant overdispersion.

To account for this, we will try the quasi-Poisson model. Our quasi-Poisson yielded a dispersion parameter that is still much higher than it ideally should be, at 13.918.

Next, we ran a negative binomial model & got a dispersion parameter of 2.465 with much lower null and residual deviances. We ran another likelihood ratio test, and got a p-value of 0.544: thus the full model isn't much better than the null.

We got the following test results from our model training & test set evaluation (MAE).

	MAE	Residual Dev (dof)	Null Dev (dof)	Dispersion Parameter
Poisson	22.36857	595.90 (35)	663.57 (39)	17.025
Quasi-Poisson	22.36857	595.90 (35)	663.57 (39)	13.918
Negative Binomial	22.39681	44.39 (35)	47.60 (39)	2.465
Negative Binomial (Reduced)	22.37500	44.40 (35)	47.45 (39)	2.456

In each of our models, the only significant predictor based on our p-values was SMR (indicating if the lightning counts were measured in the summer). We re-run the negative binomial & other models with only SMR as a predictor. None of the models experienced significant improvement, but there was slight improvement in negative binomial.

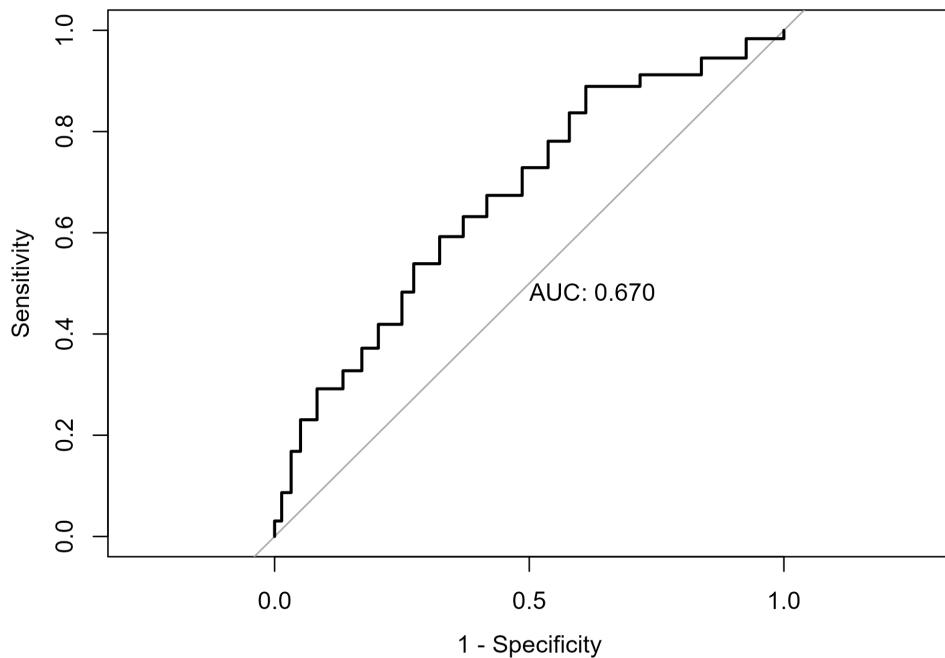
To select the best fit, we compared the descriptive statistics of the reduced negative binomial and the remaining models. The range of MAE is very small & is not a good comparison parameter.

Since the reduced negative binomial model had the lowest dispersion parameter and lowest deviance, it is the best explanatory model for the distribution of lightning count.

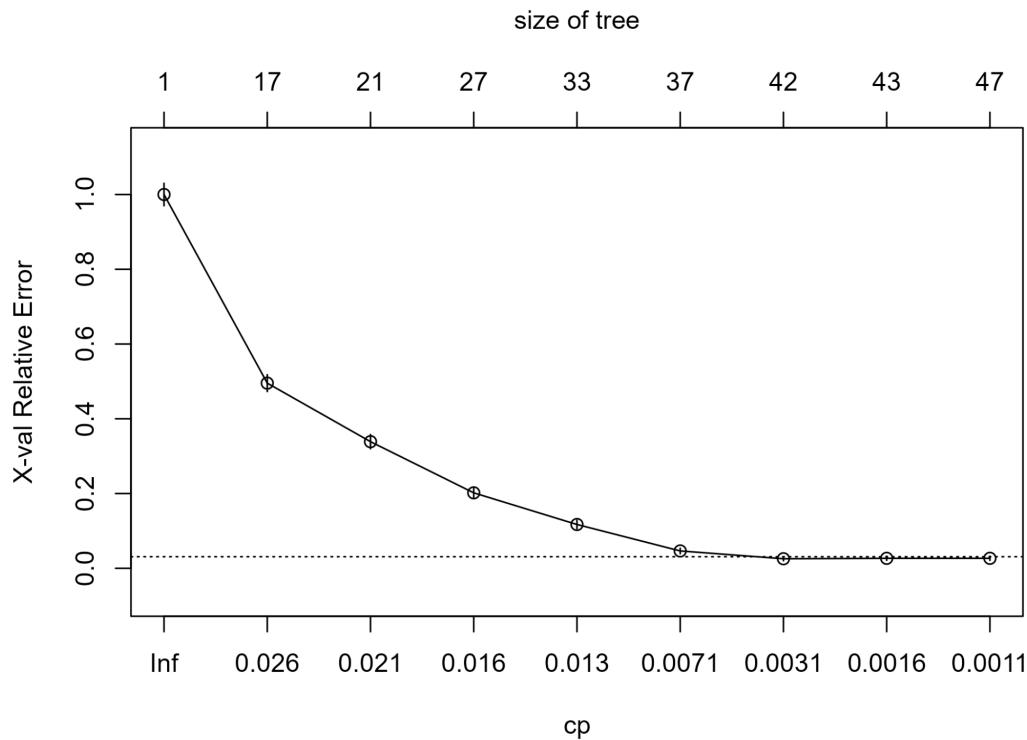
Q3::

In question 3, we have 5000 rows with three predictor columns and one binary response column. We first evaluate the distribution of 1's and 0's. We have 1078 zeroes and 3922 ones. Since we have a good amount of data, we performed an 80-20 split. We create histograms and boxplots of X1-X3 to observe mostly uniform distributions across all three [Fig. 4].

First, we tried fitting a binary logit model. From fitting the full model, we made some adjustments & used stepwise selection to find the best accuracy + sensitivity/specificity tradeoff. We plotted our ROC curve and observed a line that bows slightly to the top left corner, indicating a fit that could use improvement. We will try for a better classification using decision trees.



First, we tried CART. The tree yielded 9 CP iterations, resulting in a minimum xerror=0.031 at cp=0.00116.



We got very good accuracy = 99.4%, but we will also try random forest & XGBoost, as we may be able to improve this result.

When we did random forest, we ran 500 trees and got accuracy = 99.7%. We computed the variable importance scores and computed evaluation metrics.

We added in XGBoost as an additional model to test & got the same results as random forest. I also tried SVM. It would originally only predict ones due to a high proportion of ones. After some adjustments to gamma and scaling the training data, SVM became operable, but still yielded poor results compared to the decision trees. The following were our test set results from resulting confusion matrices:

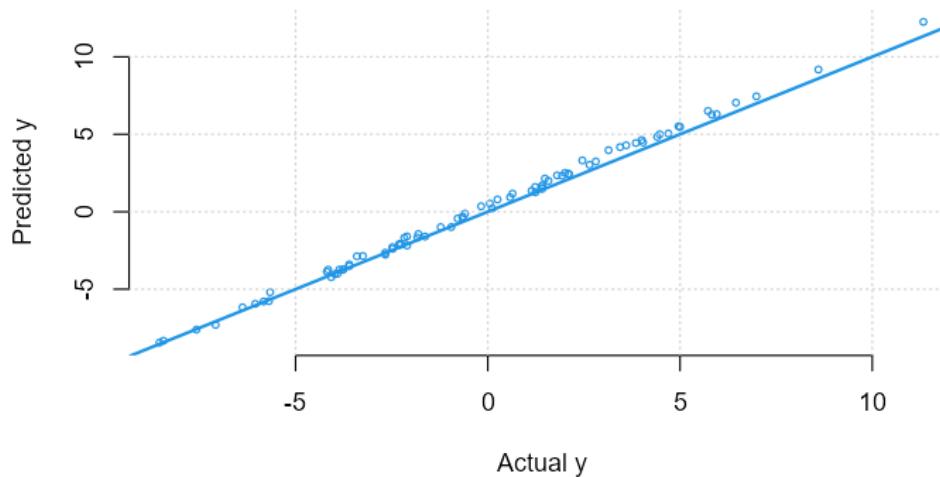
	Accuracy	Sensitivity	Specificity
Full Logit	.760	.870	.361
Stepwise Logit Function	.770	.875	.389
CART	.994	.996	.987
Random Forest	.998	.999	.991
XGBoost	.998	.999	.991
SVM	.549	.863	.283

Interestingly, we observed that X1 is by far the most important factor in determining our binary value, as it outranks the importance of both X2 and X3 by about 100x (X1 = 1335.4, X2 = 8.5, X3 = 7.9).

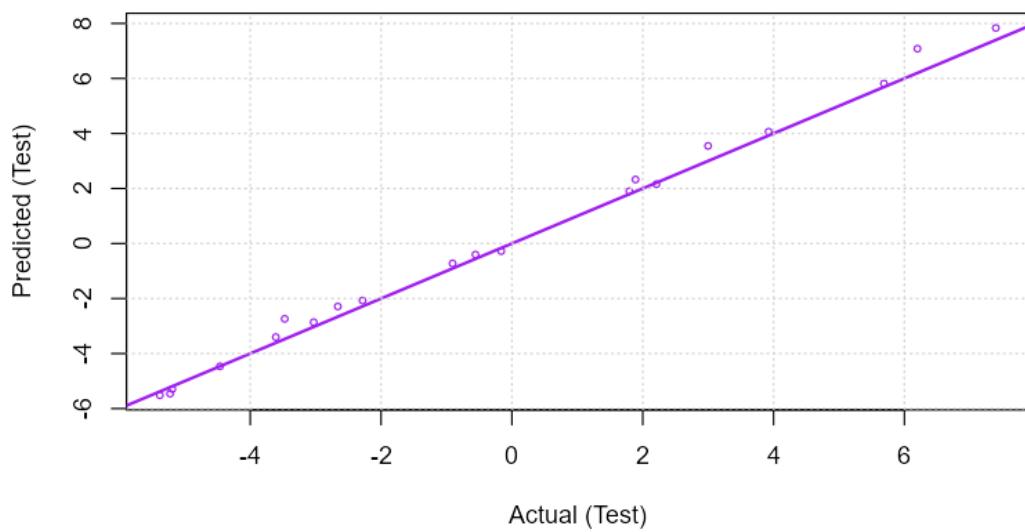
Random forest and XGBoost both yielded the best results based on our resulting confusion matrices. Since RF is generally a more stable model and ties for the best classifier, we will use it for this dataset. Accuracy is already 99.8% percent, so there isn't much room for improvement to begin with.

APPENDIX:

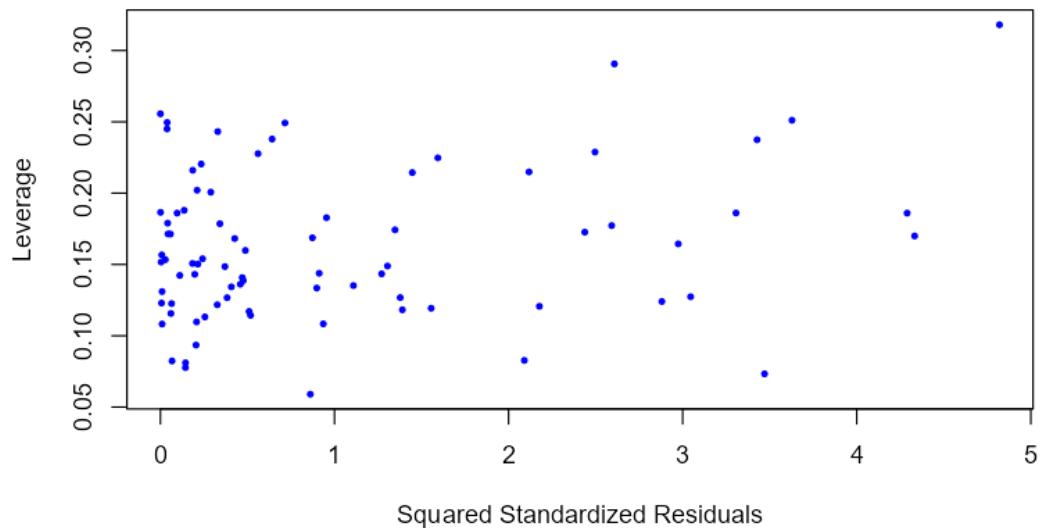
[1]



[2]



[3]



[4]

