BST 222 Final Report

Zach Clement, Nona Jiang, Dan Nolte, Willow Duffell, Addison McGhee

11/30/2021

Comparing LASSO to Other Regression Methods in the Presence of Sparsity and Small Effect Sizes

I) Introduction

The purpose of this study is to examine the impacts of a phenomenon in regression analysis and prediction known as "sparsity". In plain English, the term sparsity refers to the state of being thinly scattered or distributed. This definition has clear connections to sparsity in a regression model, which we will consider as the percent of true coefficients that are exactly equal to zero. Having high sparsity implies that the model under consideration has many predictors that are not impactful and could therefore be discarded to achieve a more parsimonious, or simple relationship with the outcome. In real-world applications, it is often advantageous to obtain a model that has fewer predictors as this allows for ease of interpretation and stronger effects from each individual covariate. However, finding such a parsimonious model is difficult in genomics research, for example, since the number of potential predictors is often very large. And many of these coefficients can be zero or very small. In these high-dimensional settings, the traditional Ordinary Least Squares (OLS) approach may not feasible if the number of predictors (p) exceeds the sample size (n). Even if $n < p$, OLS may lead to undesirable results as it will always estimate a coefficient for a variable, regardless of whether the beta is zero or not. This is also true for Ridge regression since the ridge penalty does not reduce the magnitude of coefficients to be exactly zero, preventing ineffective variables from being removed from the model.

The above example makes it clear that a more powerful technique is warranted to better estimate our parameters of interest while still maintaining a degree of parsimony in the model structure. In 1996, Stanford Statistics Professor Robert Tibishirani published a paper titled "Regression Shrinkage and Selection via the LASSO", which presented the Least Absolute Shrinkage and Selection Operator (LASSO) as an alternative to the existing regularization options. The LASSO introduces a new form of regularization that effectively "shrinks" estimates for nil coefficients to be exactly zero (Tibshirani, 1996). Fifteen-year later, Tibishirani's Stanford colleagues, Professors Trevor Hastie and Hui Zou, developed the Elastic-Net selection procedure as a way to strike a balance between Ridge regression and LASSO (Zou & Hastie, 2005).

With the motivation given, we will now turn attention to our underlying research questions:

1) How does varying sparsity impact the MSE of LASSO relative to OLS, Ridge, and Elastic-Net Regression?

2) How does LASSO perform relative to OLS, Ridge, and Elastic-Net Regression for small, non-zero beta coefficients?

3) What effect does varying the sample size and number of simulations have on LASSO and the other methods?

2) Methods

a) Models We will now present the main statistical methods that were utilized in our investigation. In order to examine the effects of sparsity in linear models, we looked at four distinct model methods that deal with sparsity in different ways. These four methods are:

- Ordinary Least Squares
- The Least Absolute Shrinkage and Selection Operator (LASSO)
- Ridge Regression
- Elastic-Net

No Penalization Ordinary Least Squares

Ordinary Least Squares, or OLS, is a method in linear regression that chooses parameters of a linear model based on a set of explanatory variables based on the principle of least squares. The Least Squares Estimation process is based on the process of minimizing the distance between the actual data responses and the fitted values/residuals of the estimated linear regression line. The line produced by the method of least squares is shown below: Ordinary Least Squares, or OLS, is a method in linear regression that estimates parameters of a linear model based on a set of explanatory variables based on the principle of least squares. A model with p explanatory variables would be modeled as such:

$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon$

Where Y is the dependent variable, β_0 is the intercept, X_j is the jth explanatory variable and ϵ is the random error term.

Least squares, also known as the minimum square error (SSE), goes about an estimation process that is based on minimizing the sum of square distances between the true data responses and the predicted values. That is, we minimize the *residuals*, which can be defined as $Y_i - \hat{Y}_i = e_i$. In order to accomplish this goal, the method of least squares produces a line such that

$Y_i - \hat{Y}_i = Y_i - (\hat{\beta}_0 + \hat{B}_1 X_i) = e_i$

is minimized over all observations by summing up all the squares of the residuals such that the sum $e_1^2 + e_2^2 + \cdots + e_n^2$ is minimized.

In other words, the ordinary least squares method chooses estimates so that the regression model deviates the least from the data, with no penalty term.

Things to note: The outputs of regression for the OLS estimator are unbiased estimators of the different β terms. According to the Guass-Markov Theorem, each of the $\hat{\beta}$ variables are unbiased and have minimum variance among all unbiased linear estimators for β. However this estimator will work best with data with no sparsity and large β terms. When we look towards models with more sparsity, OLS may not be the best method to use.

Penalization Next, we look at other methods that include penalty terms in order to create the most optimal model. These penalization methods have large benefits that allow for regression models to have the best predictive power by either minimizing the number of covariates or minimizing the weight of the covariates. When there are a large number of covariates in a data set, there may be too many to include in a regression model or not all of the covariates may be the best predictors of the regression. The regression methods below offer different penalization/shrinkage for variable selection below. However, there are some disadvantages to these methods. One being the added difficulty of model interpretation and the largest disadvantage being the presence of bias in exchange for the better predictive model.

The idea of the penalization methods is that conventional regression is first performed and then the constraint is applied to the model depending on the chosen variable selection model (LASSO, Ridge Regression or Elastic Net).

LASSO

LASSO regression, also known as Least Absolute Shrinkage and Selection Operator is a type of linear regression that uses shrinkage in a model as penalization. The main goal in LASSO regression is to select and eliminate variables by decreasing the beta coefficients to zero. If a variable is not strongly associated with the outcome, the beta coefficent is decreased to zero and is therefore taken out of the model. This regression model can also be seen as a model selection method since the model takes out what is deems as "insignificant predictors".

LASSO regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. In other words, L1 regularization adopts the constraint that the sum of the absolute-valued regression coefficients must be less than some threshold. The goal of LASSO regression is to minimize:

$\sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_p X_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

which is the same as minimizing the sum of squares. The tuning parameter λ, (either positive or 0) controls the strength of the L1 penalty and therefore the amount of shrinkage in the model. So when $\lambda = 0$, LASSO will equal the LSE. However, with this tuning parameter comes a balance of bias and variance. As λ increases the bias of the model will increase, but when λ decreases, the variance of the model will increase. Since the goal is to have the best predictive model with the least amount of bias and variance, that must be taken into account when choosing the tuning parameter if LASSO regression is chosen as the regression model method.

Things to note: The LASSO method is the preferred method when working with relatively small number of predictors that have a have a substantial effect, and the remaining predictors have coefficients that are very small or to equal zero. Therefore this method will be much more effective with a model with more zero's or sparsity. LASSO regression has many benefits that other methods don't have. This method results in a much simpler model that is much easier to interpret and it removes collinear/multicollinear variables by keeping only one of the correlated terms and eliminating the others. However, in certain settings LASSO regression may be too harsh of a method and may reduce the model by estimating more coefficients than wanted, resulting in a oversimplified model with lost information.

Ridge Regression

Ridge Regression is yet another regression model with a penalization element, but it more similar to Least Squares. Ridge regression estimates coefficients of the model by minimizing the SSE but also constrains the sums of squared coeffiencts. This method incorporates what is known as the L2 penalty which contains the constraint that the sum of the p^2 regression coefficients must be less than some threshold. This L2 penalty is used to minimize the SSE, as seen below:

$\sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_p X_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$

λ here again is the tuning parameter/shrinking penalty and will shrink the estimates of β towards to zero, but will not actually ever set a β coefficient to zero. The largest advantage Ride Regression has over Least Squares is in the bias-variance trade-off. As the tuning parameter, λ, increases, the greater the shrinkage in the model, the lesser amount of flexibility since the β coefficients are going towards zero in a somewhat proportional manner. Therefore, we have decreased variance, but increased variance. On the other hand, as λ decreases, estimated β coefficients are closer to those from OLS method which leads to less bias but more variance in these estimates.

Things to note: Ridge Regression has a lot of factors that allow it to work better in specific model cases. Ridge regression will work best when the least squared estimates have high variance, potentially when small changes in the data causes a large change in the least squared estimate. Another instance where ridge regression works well is when the number of covariates is about the same as the number of observations in the data set. Furthermore, this model is preferred when all or most of the predictors are important variables when predicting the outcome, since all the variables will be used in the model. While there are several instances where ridge regression would be the preferred method there are some disadvantages with the model. First, while the penalty term in ridge regression will shrink coefficient estimates towards zero, it will not set any of those estimates to zero. Therefore the final model includes all of the initial predictors

which could potentially cause issues with high dimensionality, parsimony, interpretation and the inclusion on collinear terms.

Elastic Net

Due to criticisms of the other penalization methods, elastic net regression was created as a happy medium between ridge regression and LASSO regression. This solution incorporates the combination of the penalties of ridge regression and LASSO in order to get the best of both models. The penalty term for ridge regression had a λ value, for its overall penalty, and an α value which balances the weight of the L1 and L2 penalties. Elastic net regression minimizes the following loss function:

$\frac{1}{2n}\sum_{i=1}^{n}\left(Y_i - X_i\hat{\beta}\right)^2 + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p}|\hat{\beta}_j|\right)$

where α is an additional tuning parameter such that:

- if $\alpha = 0$, the regression approach is identical to ridge regression
- if $\alpha = 1$, the regression approach is identical to LASSO regression
- if $\alpha \in (0, 1)$, the regression approach is Elastic Net regression

Things to note: Elastic net incorporates the best of both methods. This approach is not necessarily the best choice all the time, it depends on the given data and covariates. This approach will work best with variables with a medium amount of sparsity

b) Metrics Mean Square Error (MSE) and Variance of Regression Coefficients

In order to evaluate the performance of our simulations at predicting the true β values, we calculated mean squared error as $\frac{\sum_{i=1}^{n}(\beta_i - \hat{\beta}_i)^2}{n}$ for each β_i, when n represents the number of simulations in this context. Likewise, we calculated the variance of the β estimates across all simulations using this formula: $\frac{\sum_{i=1}^{n}(\beta_i - \bar{\beta}_i)^2}{n}$

Out of Sample MSE

To calculate out-of-sample MSE, we take each model fit and compare true Y value to predicted $\hat{Y}$ values, and compute $\frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n}$, in the typical way. However, crucially, we do not calculate this MSE on the same data that was used to fit the model. Instead, we generate a new random data sample, call it test data, and calculate associated true Y values using this new sample and the true beta coefficients. Then we used the **test data** and **modeled coefficients** from the original sample to generate predicted $\hat{Y}$ values. Then we evaluate prediction error on the test data, instead of the data that was itself used to generate the β estimates. We do this because we cannot accurately assess prediction error by testing our model on our training set. For example, the OLS model is constructed to select the β estimates which will yield the minimum squared error. This means our MSE will be as small as possible in the OLS model - but this says nothing about how this model can be used for inference with a new dataset. Out-of-sample MSE avoids this problems and allows us to make a fair model comparison.

c) Design of Simulation Study As discussed above, through this study we were interested in demonstrating the superiority of LASSO regression in those contexts with a high level of sparsity. We then consider other contexts where Ridge Regression, Elastic Net, and OLS are expected to outperform LASSO, and test our hypotheses using simulations.

Step 1: Simulation set up and data generation We begin with a basic example: we contrive a scenario where we would like to predict Y given a set of 10 covariates using the linear model below.

$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + ... + \beta_{10} x_{10}$

We then set the true parameters as follows:

$Y = 2 + 1x_1 + 0.5x_2 - 0.5x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10}$

Note that 7 of the parameters are set to 0; we made this choice because we are primarily interested in comparing model performance in the presence of sparsity.

Next, we generate a random dataset, and then run our 4 models of interest (Unpenalized OLS, LASSO, Ridge, Elastic Net) on this dataset to estimate each of the 11 β_i values using each model. We then repeat this simulation and fit regression coefficients for each model 500 times, and compare the performance of the four models. Below we explain this process in greater detail:

Step 1a: Generating random data

In order to generate our random dataset we choose a sample size of 300, and sample from a standard normal distribution for every coviariate value for each of the 300 observations. Again, every single value of every covariate is sampled from the exact same standard normal distribution. The head of this data frame looks like this:

```
##           X1         X2         X3         X4         X5         X6         X7
## 1  0.9333483 -1.8817254 -1.2325925 -0.8597954 -0.2754247  0.3497121 -0.8294332
## 2  1.8621268  1.8595435  0.1283189 -0.6511215 -1.0884418  1.6005500 -0.3417231
## 3 -0.6593261 -0.2942941 -1.0788783 -1.0841144 -0.7185210 -0.1672804  0.6119128
## 4  0.7476422 -0.4112503  1.3709654 -0.3853801 -1.8813300  0.8063871  1.1732778
## 5 -1.1407644 -0.4843944 -0.7878325  0.1158195  0.3773492 -0.9321204 -1.3256796
## 6  0.8024370 -1.2843816 -0.2175441 -1.2845680 -1.4494090  0.8327544 -0.7476742
##           X8         X9        X10
## 1  1.2316335  0.1091207  0.7982309
## 2 -0.3268431  0.7953983 -0.1739415
## 3 -0.8177599  0.4879873 -0.2044779
## 4  0.2607036 -1.9473232 -1.4173453
## 5 -0.4664293  0.2594931  1.0431931
## 6 -0.7962013 -1.0330048  1.9928137
```

Using these randomly generated covariate values, we generate corresponding Y values given our choice of parameters by simply multiplying the design matrix corresponding to our random data by the vector of parameter values, **and then adding an error term** (as drawn from the standard normal distribution). We must add an error term as per the specification of the linear model, and because otherwise OLS would exactly predict our parameter values in each simulation. Doing so we retrieve a single dataset of randomly drawn covariate values and their corresponding Y values given our choice of parameters. The head of this data frame looks like this:

```
##           Y         X1         X2         X3         X4         X5         X6
## 1  3.6552021  0.9333483 -1.8817254 -1.2325925 -0.8597954 -0.2754247  0.3497121
## 2  5.5051112  1.8621268  1.8595435  0.1283189 -0.6511215 -1.0884418  1.6005500
## 3  2.5625223 -0.6593261 -0.2942941 -1.0788783 -1.0841144 -0.7185210 -0.1672804
## 4  2.0026681  0.7476422 -0.4112503  1.3709654 -0.3853801 -1.8813300  0.8063871
## 5 -0.3333893 -1.1407644 -0.4843944 -0.7878325  0.1158195  0.3773492 -0.9321204
## 6  1.2678032  0.8024370 -1.2843816 -0.2175441 -1.2845680 -1.4494090  0.8327544
##           X7         X8         X9        X10
## 1 -0.8294332  1.2316335  0.1091207  0.7982309
## 2 -0.3417231 -0.3268431  0.7953983 -0.1739415
## 3  0.6119128 -0.8177599  0.4879873 -0.2044779
## 4  1.1732778  0.2607036 -1.9473232 -1.4173453
## 5 -1.3256796 -0.4664293  0.2594931  1.0431931
## 6 -0.7476742 -0.7962013 -1.0330048  1.9928137
```

Step 2: Simulations Details

Once we have our dataset of predictors X_i and outcome Y, what remains is to run our four regression models of interest on this dataset to retrieve point estimates for the 11 β coefficients. After running each of the models, we examine (1) the proximity of the estimated coefficients to their true underlying values in each case, as well as (2) the proximity of the predicted $\hat{Y}$ values to their true Y, measured in Mean Squared Error.

We then complete this experiment 500 times and examine the variance of the distributions of the coefficient point estimates.

As for the details of our regression models, we'll consider them in two categories: (1) the un-penalized model (OLS), and (2) the penalization models (Ridge, LASSO, Elastic Net)

1) *Un-penalized Model (OLS)*: For this model, we simply regress Y on $X_1, X_2, ..., X_{10}$ using OLS using the below R command.

```
lm(Y ~ X1:X10, data = data)
```

2) *Penalization Models*: We used the same basic procedure for fitting each of these 3 models. For each we use the `glmnet` library in R and run a model in this form:

```
glmnet(y = data$Y, x = data[,2:ncol(data)], family = "gaussian", alpha = alpha, lambda = lambda)
```

Where we set alpha equal to 0, 0.5, and 1 for Ridge, Elastic Net, and LASSO, respectively. Of course, a more rigorous study would also attempt to select an optimal alpha level for the Elastic Net model, but choosing alpha = 0.05 for Elastic Net is sufficient for our illustrative purposes.

As for selecting the lambda level, used 5-fold cross validation across many levels of lambda and selected the lambda in each case associated with the lowest average MSE in the test set. Then we fit each penalization model with this lambda value. The R code used to retrieve the optimal lambda level is stated below:

```
cv.glmnet(y = data$Y, x = as.matrix(data[,2:ncol(data)]),
          nfolds = 5, family = "gaussian", alpha = alpha) %>% .$lambda.min
```

Step 2a: Simulation at Varying Sparsity Levels, Sample Sizes, and Coefficient Sizes

To answer the questions of how each model performs at varying levels of sparsity in the model, sample sizes, and coefficient sizes (specifically when the true parameters are non-zero but instead very small) we ran a series of additional simulations as described

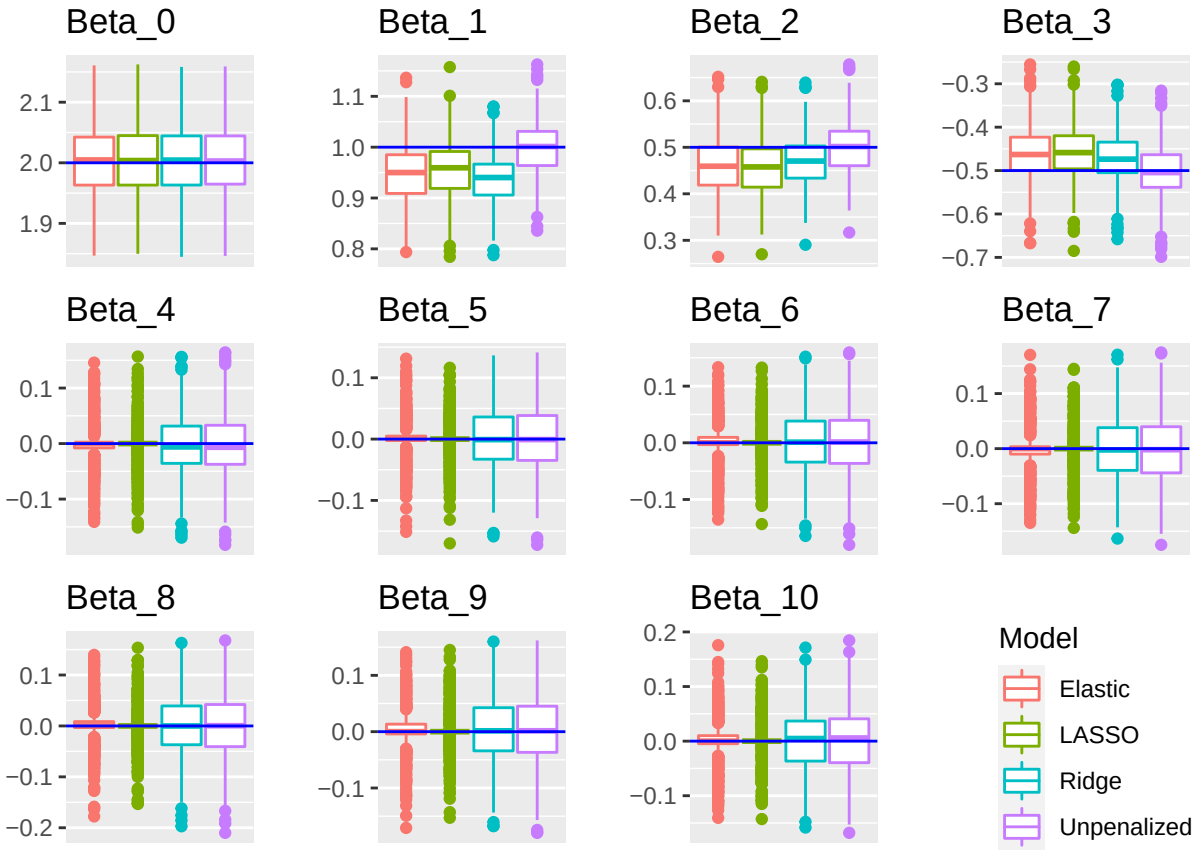
To evaluate the prediction MSE for each model under different levels of sparsity and at different sample sizes, we repeated the simulation as described above for varying sparsity levels (.1, .5, and .98), with these values representing the percentage of the true beta coefficients being zero. We also repeated the simulation for each level of sparsity at different sample sizes (200, 800, 1600). We will used 500 replications under each condition to conduct the estimation, using 100 covariates in each simulation.

To evaluate the effect of small coefficient sizes on the performance of each model, we set sparsity fixed at 0.1 and varied the size of non-zero coefficients. We set small coefficients to lie in the range 0.01-0.1, and we allow the remaining coefficients to lie in the range 0.5-1.5. For example, in the setting with small = 0.5; half of the true non-zero coefficients lie in 0.01-0.1, and half lie in 0.5-1.5.

Finally, we ran the simulation for all 9 different combinations of sparsity (0.1, 0.5, 0.98) and coefficient sizes (small = 0, 0.5, 1) as described above.

3) Results

Below are a series of boxplots which show the distribution of point estimates for each of the 11 β parameters in our original simulation Recall that for the simulation study we set $\beta_0 = 2$, $\beta_1 = 1$, $\beta_2 = 0.5$, $\beta_3 = -0.5$, and $\beta_4 = \beta_5 = ... = \beta_9 = \beta_{10} = 0$. In this setting we set n = 300, and run each model 500 times.

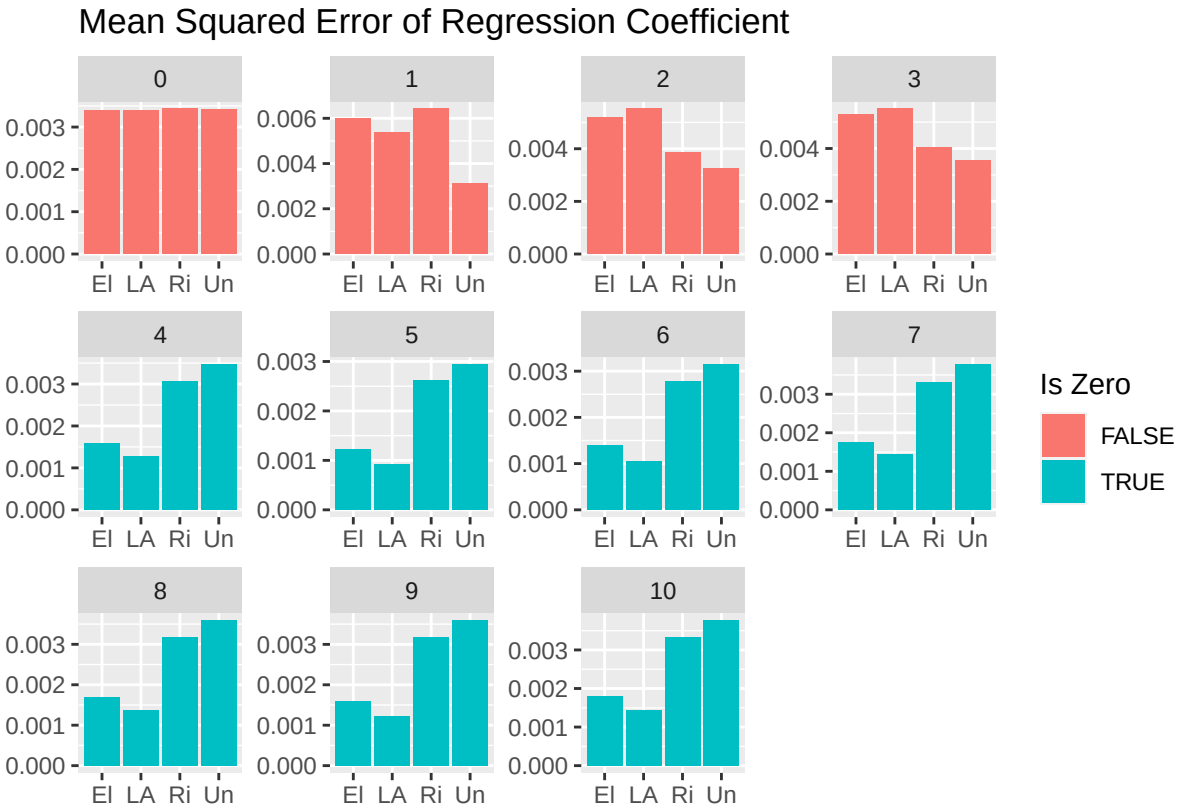


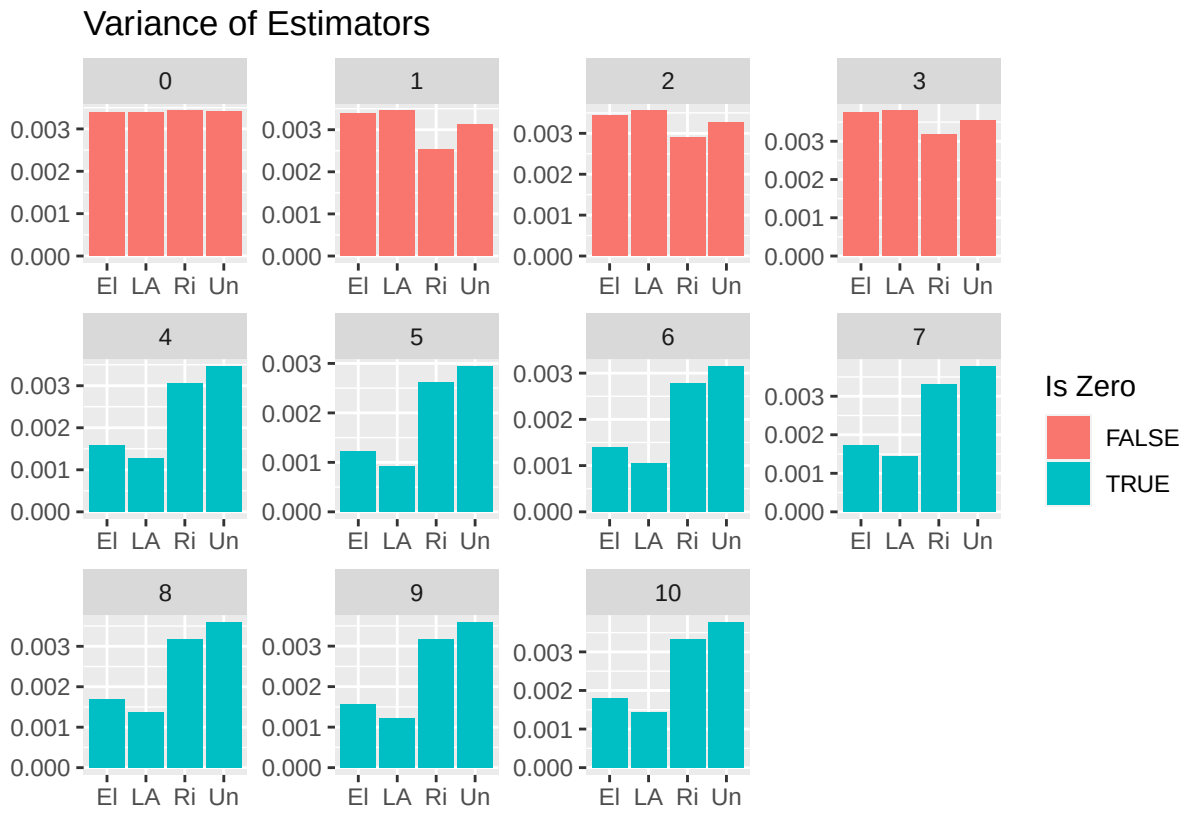
We can see from the boxplots of the beta coefficients that LASSO, Ridge, and elastic net tend to underestimate (in absolute value terms) the true parameter, while the unpenalized model is unbiased. This jibes with our expectation, as the penalization methods are shrinking the truly non-zero coefficients (often all the way to 0 in the case of LASSO), and in so doing are introducing bias in exchange for smaller variance. The clear superiority of Unpenalized OLS on predicting β_1, β_2, and β_3 should serve as a cautionary tale that these penalization methods, while powerful, certainty introduce bias when predicting relatively "large" parameter values and should be used with care.

On the other hand, when parameters are truly zero, – despite the fact that Unpeanlized OLS parameters are unbiased in these cases – the penalization methods outperform Unpenalized OLS in that they have much lower variance; in other words, they more accurately predict the true 0 betas much more frequently than do does Unpenalized OLS.

It's also important to note that in this example Ridge does not perform much better than Unpenalized OLS for the non-zero coefficients in terms of variance. This is partly due to the fact that Ridge does not send any coefficients to 0; but it's slighter smaller variance for these parameter estimates is due to the shrinkage introduced by the L1 penalty term.

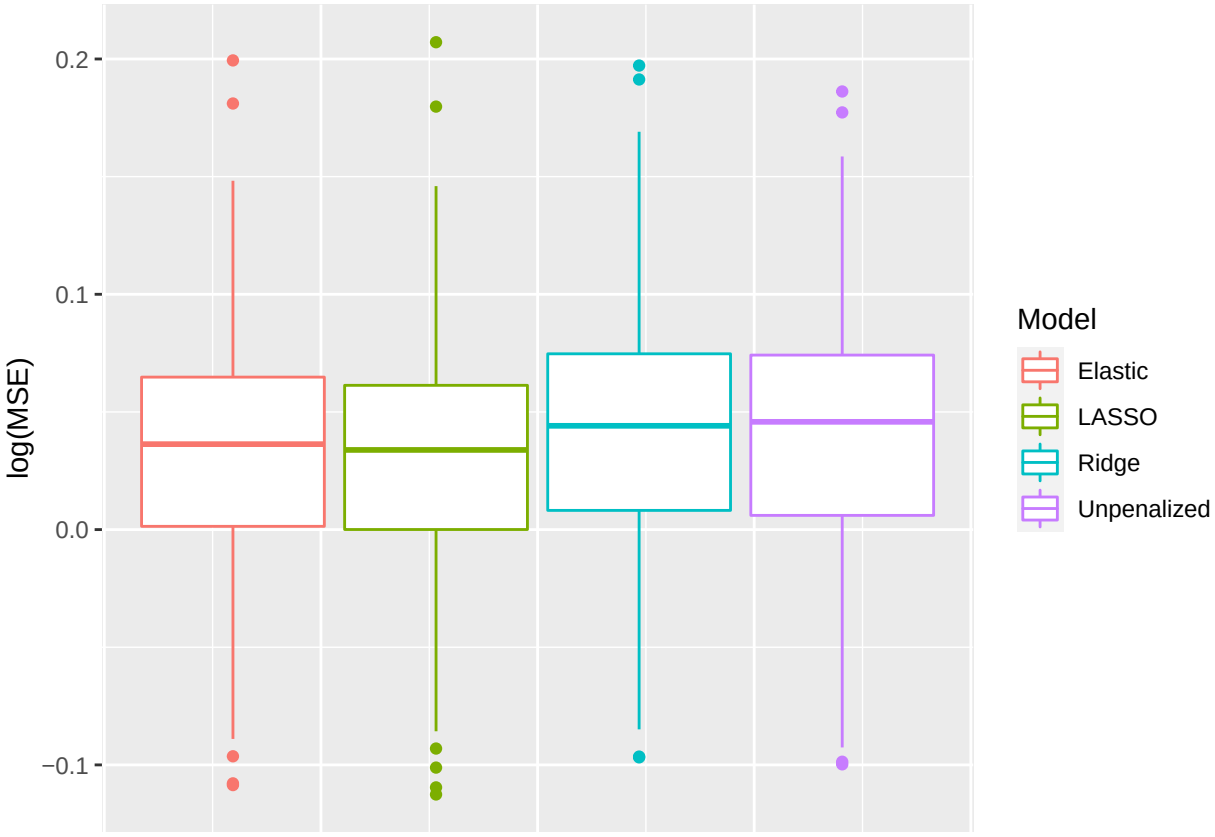
Separately, we are interested in the MSE and variance of each β. below I plot both of these:


Mean Squared Error of Regression Coefficient
0
1
2
3
4
5
6
7
8
9
10
El LA Ri Un
Is Zero
FALSE
TRUE



When evaluating the mean squared error of our estimates of regression coefficients, Elastic Net and LASSO regression tend to have lower MSE for coefficients when they are equal to zero; this finding is consistent with the boxplots above. Again, MSE is significnatly lower for truly non-zero parameters when estimated with Unpenalized OLS. As for variance, the variance in β estimates is much lower for Elastic Net and LASSO regression for zero coefficients, but is comparable to Unpenalized OLS for non-zero coefficients.

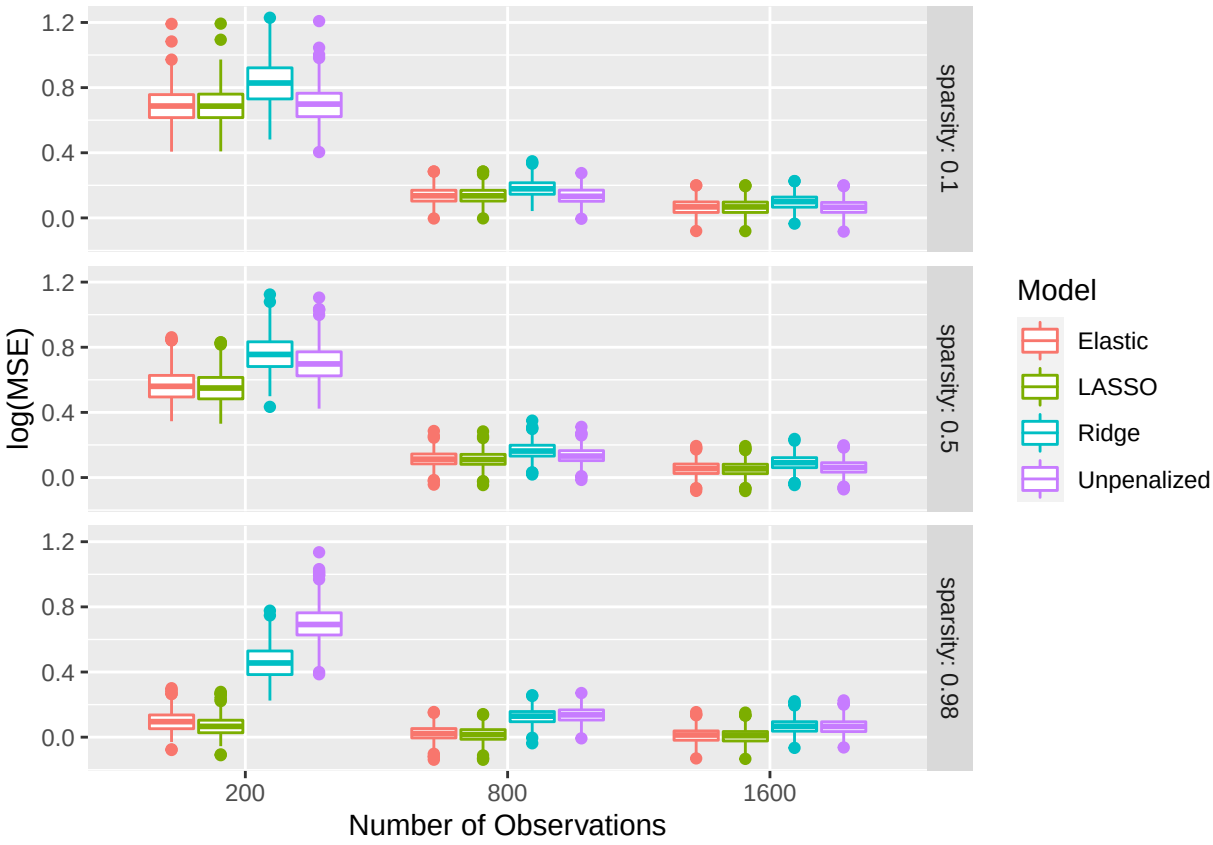
We've now reviewed the estimates of the *beta coefficients* relative to their true underlying values. A similar but related question we'd like to answer is how well each model performs at predicting the true underlying *Y values*. In order to do this, we compare models using **out-of-sample MSE** (defined above.)



It is clear that in this setting of relatively high sparsity, Elastic Net and LASSO outperform Ridge and un-penalized OLS in terms of out-of-sample MSE. This is because bias introduced by shrinking the non-zero parameters in the penalized models is only a small price to pay for the steep reduction in variance when predicting the truly 0 parameter values. In this setting where there is high sparsity, it is clear that LASSO reigns supreme.

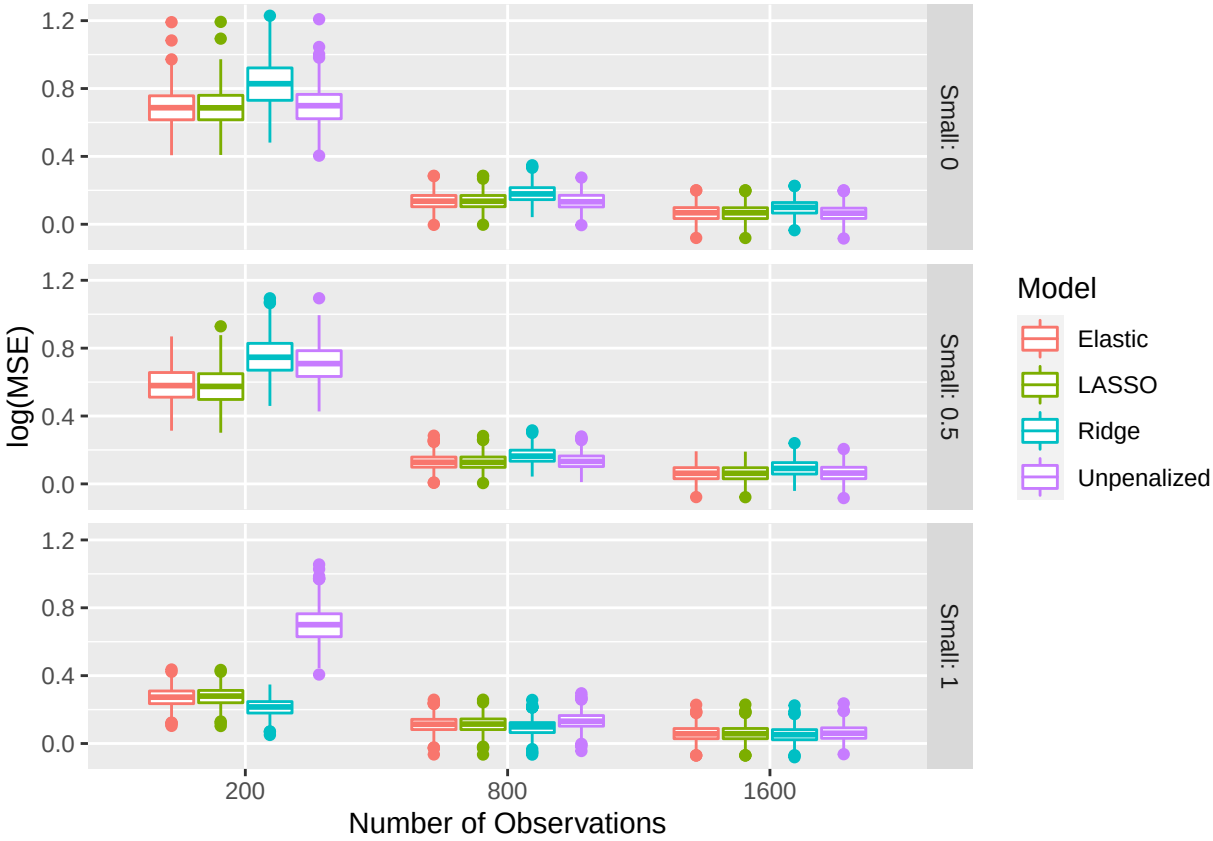
However, what if there is slightly less sparsity? Or what if the true parameters are non-zero but instead just very small? How does each model perform in these settings. We now proceed to test some of these questions to better understand the trade offs implicit in these penalization methods.

First, we will evaluate the prediction MSE for each model under different levels of sparsity (.1, .5, and .98), and at different sample sizes (200, 800, 1600). A sparsity of 0.1 means that 10% of the true beta coefficients are 0, for example. We will use 500 replications under each condition to conduct estimation, and we will use 100 covariates in each simulation.



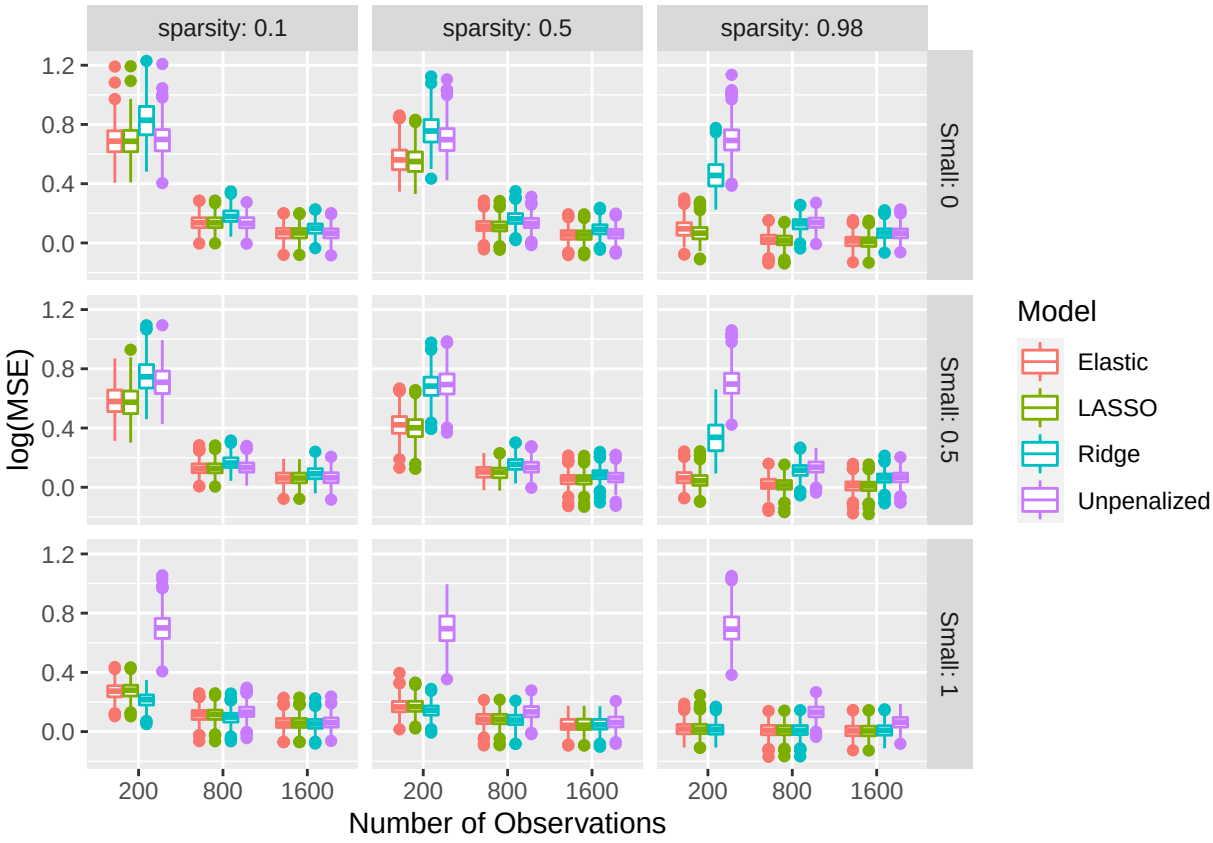
It is clear that in the very high sparsity setting (i.e., the setting where nearly all coefficients are truly 0) we see that the OLS model performs quite poorly compared to the penalization methods (especially when there are fewer observations). A somewhat counter-intuitive finding is that the penalization methods perform quite well on median in the low sparsity setting as well; this can be explained by the fact that in these settings, cross-validation will set the optimal value of λ to close to 0. A further question of interest might be to examine model performance for even lower sparsity (or even 0 sparsity), and increase the number of covariates tested from 100 to say, 200; and to compare results.

Now we consider the setting where the non-zero coefficients are close to 0, (i.e., "small"). In this setting, we fix sparsity at 0.1 (i.e., 10 of the 100 parameters are truly 0) and vary the size of non-zero coefficients. We set small coefficients to lie in the range 0.01-0.1, an we allow the remaining coefficients to lie in the range 0.5-1.5. So for example, in the setting with `small = 0.5`; half of the true non-zero coefficients lie in 0.01-0.1, and half lie in 0.5-1.5. In this setting, we expect that severe penalization methods like LASSO and Elastic Net will send too many of these coefficients to 0, and the OLS model will overestimate these small true β values.



Indeed, we see that Ridge regression is the highest performing model in the setting of high "smallness", and this is especially true with a lower number of observations. However, as sparsity increases, LASSO and Elastic Net begin to perform as well or better than Ridge Regression.

Below we present results for all 9 combinations of sparsity and coefficient size.



Here we see that OLS performs about as well as Elastic Net and LASSO in the low sparsity, low smallness setting, and OLS has especially poor performance, relatively speaking, in the high smallness setting. In each case, higher sample sizes lead to convergence of MSE across the four models.

4) Summary/Conclusion

In conclusion, in our study we ran several simulations to answer the questions of how sparsity impacts the MSE of unpenalized OLS versus penalized (LASSO, Ridge, Elastic Net) models. Our overarching finding is that in settings of high sparsity, penalized methods outperform the traditional unpenalized OLS, with LASSO performing best. Our project highlights the utility of using LASSO under situations of high sparsity. Many fields in biology and medicine have many predictors that are often highly correlated including but not limited to studies in genomics, cell signaling, and immunology. Thus, identifying a model that is able to account for situations of high sparsity is paramount and has many implications for biological research.

We found that with non-zero beta coefficients, the penalized models underestimate the true parameter while the unpenalized OLS remains unbiased. This finding highlights the idea that while penalization methods such as LASSO, Ridge, and Elastic Net are powerful in the correct setting, they do introduce bias when predicting relatively "large" parameter values and should be used appropriately. Conversely, for true zero betas, the penalization methods outperformed the unpenalized OLS in that these models displayed lower variance and more accurately predicted the true zero betas with Elastic Net and LASSO having the lowest MSE for the coefficients.

Furthermore, in settings of high sparsity, Elastic Net and LASSO also outperform the other two models in terms of out-of-sample MSE. In these settings, the bias introduced by shrinking the non-zero parameters in the penalized models becomes a relatively small price to pay for the steep reduction in variance when predicting the truly zero parameter values. In these settings of high sparsity is where the merits of LASSO truly shine. These findings are recapitulated when a series of simulations were repeated with varying degrees

of sparsity and sample sizes. And when considering a setting of high "smallness," we find that Ridge regression outperforms the other models. Finally, we observe a convergence in MSE for each of the 4 models (no matter the sparsity and smallness conditions) as sample size increases.

Together our simulation findings suggest that in settings of high sparsity , LASSO is a powerful way to obtain a parsimonious model. Our findings highlight the ability of LASSO to be applied to many fields, for example biology and medicine, where the number of predictors is large relative to the sample size.

5) Acknowledgments

We would like to acknowledge the entire BST 222 Fall 2021 teaching team at Harvard T.H. Chan School of Public Health. In particular, we would like to thank Professor Rui Duan for all her wonderful teaching and the opportunity to perform and learn from this simulation study. We extend special appreciation to Larry Han who has provided not only support throughout the BST 222 course, but also took time to offer invaluable advice and mentorship to our team on this project.

6) References

- "Ordinary Least Squares Regression (OLS)." XLSTAT by Addinsoft, 2021, https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols.
- "Images and Materials copyright BST 210 HSPH, Erin Lake, and Harvard University"
- https://www.statisticshowto.com/lasso-regression/
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. J. R. Statist. Soc. B, 58, 267–288.
- Zou, H., Hastie, T. (2005). Regularization and variable selection via the elastic net. J. R. Statist. Soc. B, 67(2), 301–320.