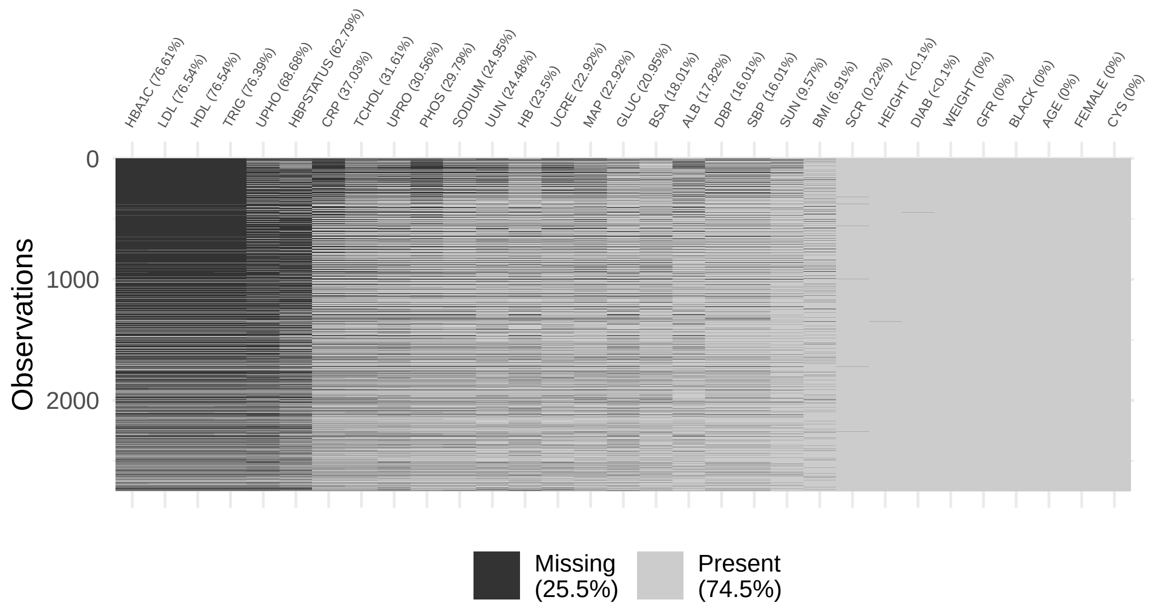
PDA Assignment #5

Anthony Sisti

11/18/2019

Preliminary Analysis

The first problem requires that we work with the 'iod' data set to predict GFR values. After selecting the factors that were specified on the assignment, we have 2749 observations of 31 variables. Since we are using the data to predict GFR, we examine how much of it is missing.



The visualization above shows over 70% of missing data for some variables, and complete data for otheres. When considering the entry values for all variables within a row simultaniously, we only have approximately 20% complete observations. Given this small number, we investigate the summary statistics of the variables currently in the data set. We display the summary statistics for variables which have over 1000 missing values

	UPHO	HBA1C	TRIG	LDL	HDL	CRP	HBPSTATUS
Mean	849.88	5.75	173.85	147.15	39.53	0.61	0.7
Stdev	346.74	0.95	128.42	41.57	14.01	0.83	0.46
Minimum	96	3.9	25	44	14	0	0
Q1	610	5.3	92	119	29	0.15	0
Median	818.53	5.6	140	143	37	0.34	1
Q3	1026	6	213	172	47	0.77	1
Maximum	4797.7	15	1420	301	110	10.7	1
NAs	1888	2106	2100	2104	2104	1018	1726

Were these seven variables to be removed, we would have 57% complete cases, up from 20%. Due to this, and the fact that each of these variables have at least 37% of their observations missing, we decide to omit them.

We now consider the cases that remain incomplete even after the removal of these seven varables (1187/2749 cases). We assess any major differences between the incomplete cases and the complete cases, and after examining the summary statistics for each remaining factor, three variables appear to vary widely between the two groups. We display their summary statistics below.

Table 1: Summary of Select Variables from Complete Cases

	UUN	UCRE	DIAB
Mean	8.98	1587.93	0.02
Stdev	3.59	581.57	0.14
Median	8.6	1525.54	0
Minimum	0	247.38	0
Q1	6.38	1154.39	0
Q3	11.07	1951.72	0
Maximum	32.03	4025.6	1
NAs	0	0	0

Table 2: Summary of Select Variables from Incomplete Cases

	UUN	UCRE	DIAB
Mean	2212.7	920.89	0.27
Stdev	4124.83	856.45	0.44
Median	111.5	1015	0
Minimum	2.71	0.9	0
Q1	10.85	5.6	0
Q3	366	1548	1
Maximum	20385	6199	1
NAs	673	630	1

The most striking difference is found for the ‘UUN’ variable. While the minimum and first quarter don’t show extreme varition, the rest of the statistics appear extremely inflated in the ‘Incomplete Cases’ group. Since the typical values for Urine Urea Nitrogen (UUN) fall between 8 and 25g (healthline.com), it is likely that there was some data collection error within the removed cases group. The discrepancey in the ‘UCRE’ variable is mostly at the lower recorded levels for both groups. The incomplete cases have a minimum of 0.9 and a first quarter of 5.6. Since values typically range from 1000 to 3000 mg/24hrs (healthline.com), there might be some data collection error in the removed cases group. The diabetes variable also shows a lot of variation. Over 25% of the removed cases are diabetes cases, while only 2% of remaining cases are diabetic. This could be the reason we see the extreme differences in the other two variables but our domain knowledge is limited, so it is difficult to know. We decide the best course of action is to keep only the complete cases, but remove the diabetes varible since it only appears as a positive indicator 2% of the time in these complete cases. We also ignore the multicolliniarity between variables because we know we are using model selection procedures that should siphen out predictors that do not improve the effectiveness of the model. The final data set consists of 1562 complete observations of 23 variables. The final set of variables used for modeling are shown below.

- WEIGHT: (kg)
- BMI: body mass index
- GFR: glomerular filtration rate
- UCRE: urine creatinine
- UUN: urine urea nitrogen

- SUN: serum urea nitrogen
- SCR: serum creatinine
- TCHOL: total cholesterol
- ALB: albumin
- PHOS: serum phosphorus
- HB: hemoglobin
- MAP: mean arterial pressure
- UPRO: urine protein
- BSA: body surface area
- SODIUM: sodium
- GLUC: glucose
- BLACK: black race
- HEIGHT: height (cm)
- AGE: age
- FEMALE: female
- CYS: serum cystatin
- DBP: diastolic blood pressure

Problem 1

Write a bootstrap algorithm to adjust for optimism on training data in using forward and backward stepwise regression to select a best model for predicting GFR.

(a) Find forward and backward regression to the entire dataset and compute R^2.

Using the reduced data set discussed in our preliminary data analysis, we fit forward and backward stepwise regression with AIC as the selection criterion. The model formulas for the 'best' models determined by both methods are shown below.

Backward Selection Model:

```
GFR ~ WEIGHT + BMI + UCRE + SUN + SCR + ALB + PHOS + UPRO + GLUC + BLACK + AGE + FEMALE +
CYS
```

Forward Selection Model:

```
GFR ~ CYS + UCRE + BLACK + SCR + FEMALE + AGE + PHOS + UPRO + SUN + ALB + HEIGHT + GLUC
```

Note, the backward selecion model consists of thirteen predictors while the forward selection model contains twelve. The backward model contaings `WEIGHT` and `BMI` while the forward selection model contains `HEIGHT`. In case of backward selection, the R^2 value is 0.721 and in the case of forward selection, R^2 is 0.720.

To satisfy parts (b) and (c) we bootstrap the data 1000 times and fit forward and backward regression models on the sample, saving the R^2 value each time. We then calculate the R^2 on the test set that did not appear in the bootstrap by computing

$$1 - \frac{\text{Residual SS}_{test}}{\text{Total SS}_{test}}$$

To satisfy (d) and (e), we subtract $R^2_{bootTest}$ from $R^2_{bootTrain}$ for each of the 1000 bootstrap samples. This leaves us with 1000 $R^2_{bootOpt}$ values for both the forward and backward models. When we average these values across the 1000 samples we find that the mean $R^2_{bootOpt}$ is 0.017 for the backward regression , and 0.014 for the forward regression.

Our orignal R^2 values were 0.721 for the backward selection and 0.720 for forward selection. After subtracting their respective optimism measures computed from the bootstrap samples, we find an estimates for the test R^2 to be 0.704 and 0.705 for the backward and forward selection models, respectively. A summary table for this information is displayed below.

Model	R^2_{full}	Mean $R^2_{bootTrain}$	Mean $R^2_{bootTest}$	Mean $R^2_{bootOpt}$	Estimated Test R^2
Backwards	0.721	0.725	0.708	0.017	0.704
Forwards	0.720	0.724	0.710	0.014	0.706

Both models produced extremely similar results, with the full R^2 values and test R^2 estimates within 0.02. There was slightly more optimism displayed by the backwards selected model, but the difference is extrememly small.

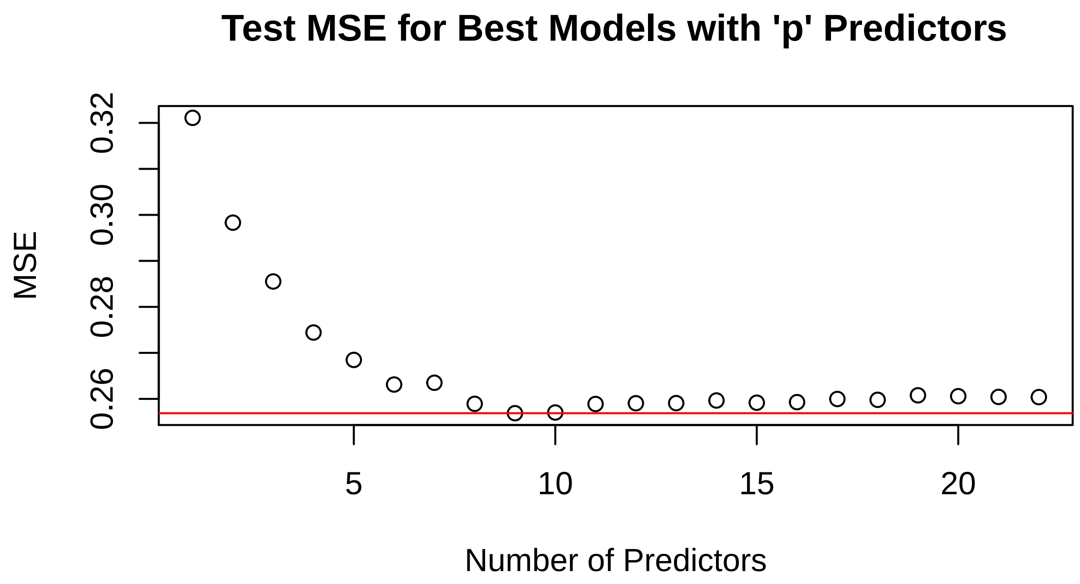
Problem 2

Use cross-validation to construct a predictive model with a) stepwise regression; b) ridge regression; c) lasso regression. At the end, refit using the best fittting model of each type on the whole dataset. Compare the estimate of test R^2 between the bootstrap approach and the stepwise, ridge and lasso approaches.

This question was interpreted as asking us to compare the test R^2 from the bootstrap method to the R^2 of the best model (as decided by cross validation) from the stepwise, ridge and lasso approaches when they are fit on the entire data set. We will first fit a stepwise model on a training data set, and fit and find the R^2 of the best model on the entire data set. This will be followed by a similar procedure for LASSO and Ridge Regression where we will use 5-fold cross validation to determine the best norm penalizaion value. We will compare these and their R^2 values on the entire dataset to eachother and the bootstrap 'test' R^2 value. Note, training and test data is standardized prior to model fitting, this will allow for the magnitude of the coefficients to be compared accros models and across predictors in problem three.

Stepwise

We write a procedure to combine stepwise regression and cross validation. The prodcedure breaks the data set into two parts: a training and a test data set. Using AIC as a selection criterion, we choose the best model with 'p' predictors ('p' ranging from 1 to 22) on the training data set. We then evaluate the performance of each of these 22 models on the test data set using mean squared error. We plot these values and choose a model that perfoms the best using this metric, while remaining at least somewhat parsiminous if the MSE values are simlar. The plot below displays how the best model for each number of predictors performed when predicting on the test data set.



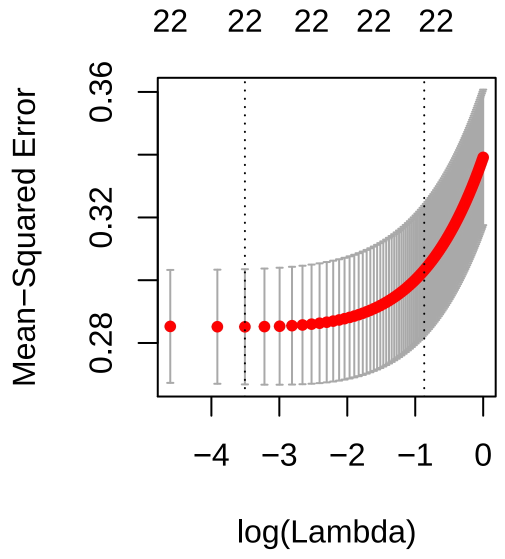
The model with nine predictors had the lowest overall MSE on the test set, and shows improvement over the models with less predictors. Therefore, we select this model to move forward and fit to the entire data set. The formula for this model is displayed below.

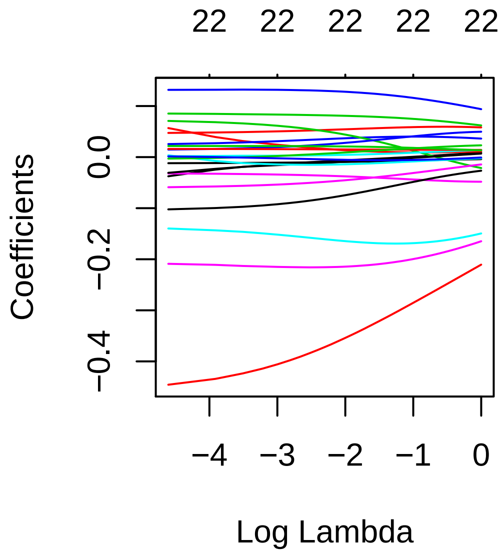
```
GFR ~ UCRE + SUN + SCR + ALB + PHOS + BLACK + AGE + FEMALE + CYS
```

The model contains nine predictors, smaller then the stepwise models fit on the entire data set in problem 1. The R^2 value when this model is fit on the entire data set is 0.719.

Ridge Regression

In order to select the best model for Ridge Regression, we use cross validation with 5 folds to evaluate different λ penalization values. We plot the average MSE across each of the five folds, letting the x-axis be the log of the lambda value used in each model. We also visualize how the penalization shrinks each of the coefficients, by fitting the model on the whole data set and showing how the magnitude of each coefficient changes as the L2 norm is forced to decrease due to larger penalization.

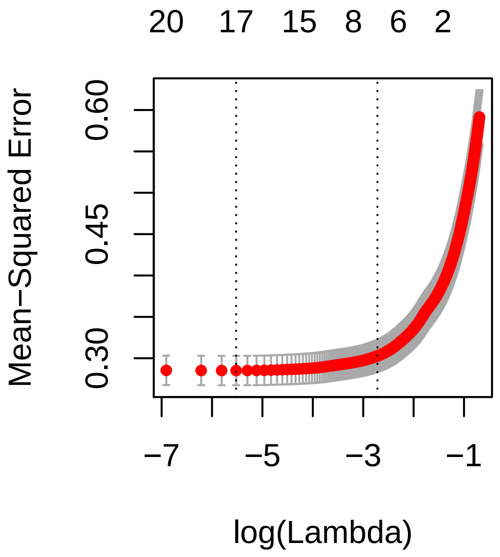


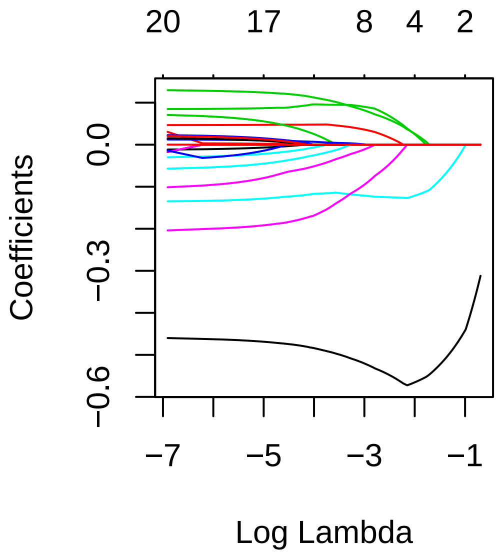


Based on the cross validation MSE, the optimal λ value to use for penalization is 0.03. We fit this model on the whole data set and compute the R^2 value, which comes out to 0.721.

LASSO

Similar to Ridge, in order to to select the best model, we use cross validation with five folds to evaluate different λ penalization values. We plot the average MSE across each of the five folds, letting the x-axis be the log of the lambda value used in each model. We also visualize how the penalization shrinks each of the coefficients, by fitting the model on the whole data set and showing how the magnitude of each coefficient changes as the L1 norm is forced to decrease due to larger penalization.





Based on the cross-validation mean squared error, the optimal lambda to use for penalization is 0.004. The MSe plot shows that this penalization leaves us with 17 predictors that have not been forced to zero. These predictors will be assessed in part three. We fit this model to the whole data set to determine the R^2 value, which comes out to be 0.721.

R^2 Comparison

Stepwise	Ridge	LASSO	Ave Test R^2
0.719	0.721	0.721	0.705

This table displays the R^2 values from each of the three models we fit above, as well as the average of the estimates of test R^2 from the forward and backward selection models fit in problem one. The R^2 values from each of the models were slightly larger than the estimate of test R^2, with the stepwise model having the smallest R^2 value of the three. The fact that the R^2 from the models is larger than the estimate for test R^2 is not surprising. The test sets that the models were optimized for was the majority of the full data set. Since the model was optimized for a majority of the observations in the full data set, we are not getting a true 'test' R^2.

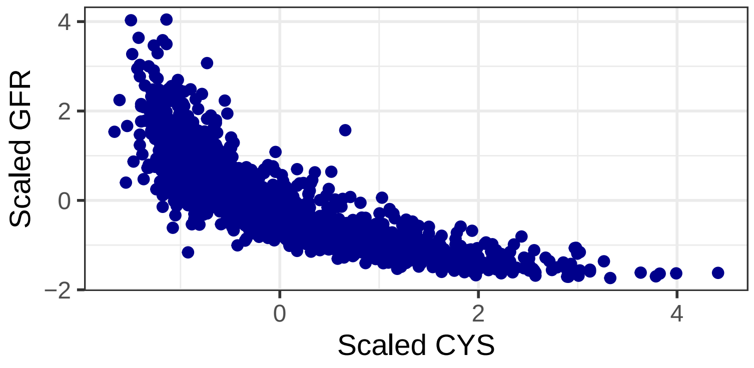
Problem 3

We summarize the model coefficeints and included predictors below. Recall, all coefficients were calculated on a standardized data frame, the models contained no intercepts.

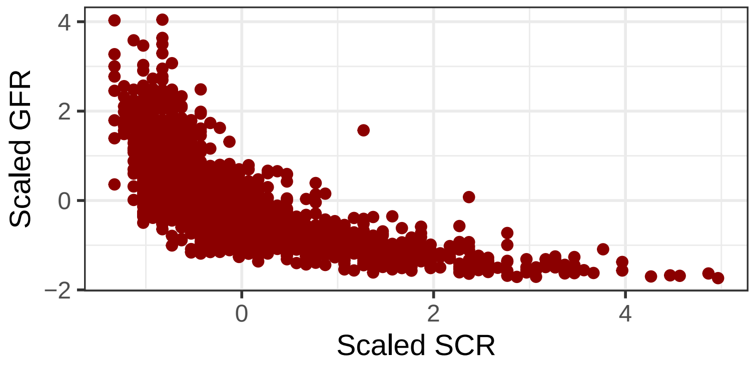
Table 3: Predictor Coefficients by Model

	Stepwise	Ridge	LASSO
WEIGHT	–	-0.0223	**0.0000**
BMI	–	0.0349	0.0029
UCRE	0.0972	0.0841	0.0859
UUN	–	0.0288	0.0187
SUN	-0.1219	-0.1468	-0.1309
SCR	-0.2266	-0.2125	-0.1975
TCHOL	–	-0.0114	-0.0091
ALB	0.0611	0.0490	0.0470
PHOS	0.0686	0.0661	0.0629
HB	–	0.0171	0.0112
MAP	–	0.0057	**0.0000**
UPRO	–	-0.0331	-0.0255
BSA	–	-0.0199	**0.0000**
SODIUM	–	0.0158	0.0122
GLUC	–	0.0218	0.0165
BLACK	0.1297	0.1319	0.1268
HEIGHT	–	-0.0097	-0.0243
AGE	-0.0460	-0.0562	-0.0506
FEMALE	-0.0850	-0.0972	-0.0898
CYS	-0.4633	-0.4239	-0.4649
DBP	–	-0.0019	**0.0000**
SBP	–	-0.0025	**0.0000**

In the table above, we find that the stepwise model is the most sparse, only containing 12 predictors. The LASSO procedure reduced the number of predictors to 17, forcing the coefficients of 5 predictors to be zero. The ridge regression contains all 22 predictors as we would expect, but four of the coefficients were reduced to the hundredths place. In both the LASSO and stepwise models, `DBP`, `SBP`, `BSA`, `MAP` and `WEIGHT`, were either not included, or had regression coefficnets of zero. `DBP`, `SBP`, `BSA` and `MAP` had the smallest regression coefficients by magnitude in the Ridge model, while `WEIGHT` had the tenth smallest. The models conclusively show that `DBP`, `SBP`, `BSA` and `MAP` were, if not the least effective, amongst the least effective predictors of GFR. Both LASSO and stepwise procedures deemed `WEIGHT` to be an ineffective predictor, while the Ridge regrssion found it to me moderately useful. The most influential predictor of `GFR`, by nearly double the coefficient magnitude in each of the models, was `CYS`. The coefficients for this variable in all three models remained between -0.424 and -0.465, while the stepwise and LASSO coefficients are the same through the tenths place. This strong, negative association is not surprising when we plot `GFR` and `CYS` against one another.



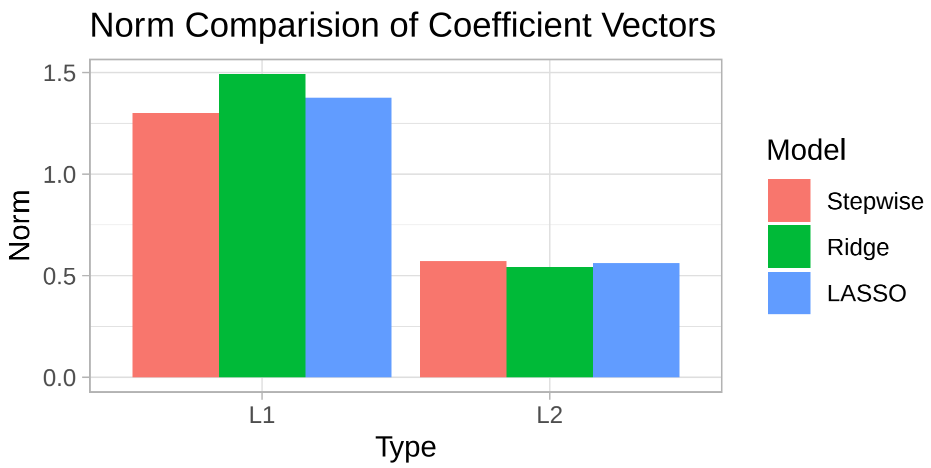
Another strong predictor, the second most influential in all three models, was `SCR`. The coefficient for this predictor ranged between -0.1975 and -0.2270. This is a significant jump in magnitude above the next largest continuous predictor coefficient found any of the models. Again, this is not surpising after examining the relationship between `GFR` and `SCR`.



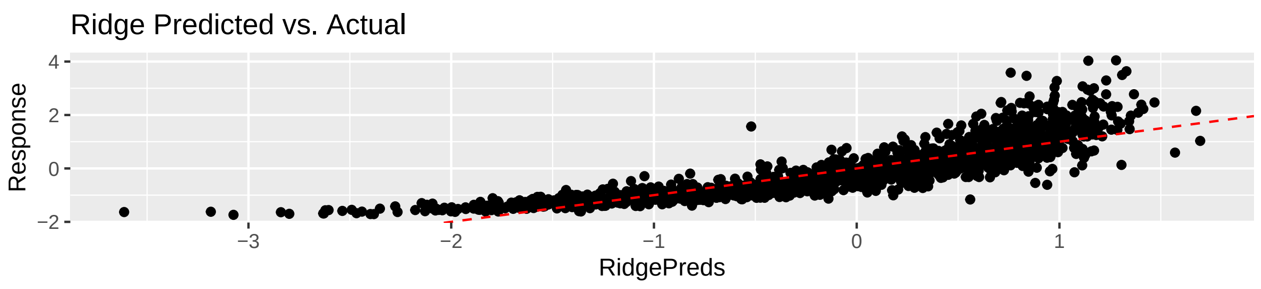
We next examine the indicator variables `FEMALE` and `BLACK`. Both of these predictors were found in all three models with coefficient magnitudes larger than 0.08, indicating that they matter for prediction of GFR. The concensus based on the coefficient outputs is that, being female decreases the expected `GFR` value by approximately 0.9 standard deviations, while being black increases the expected `GFR` value by approximately 0.13 standard deviations. In order to compare the magnitude of the coefficient vectors, we create a table and plot the L^1 and L^2 norms of each.

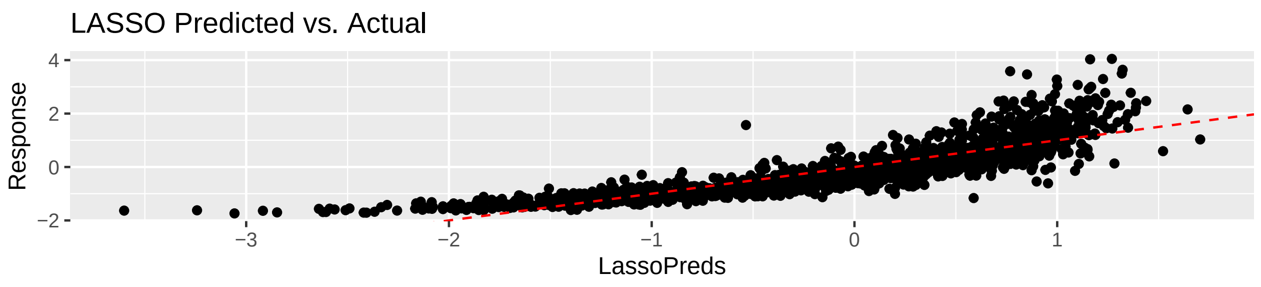
Table 4: Coefficient Vector Norm Comparison

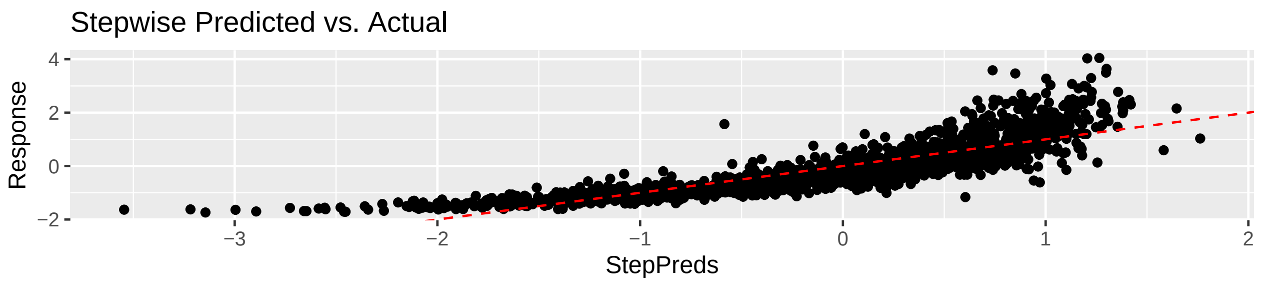
	Stepwise	Ridge	LASSO
L^2	0.57	0.544	0.561
L^1	1.30	1.493	1.377

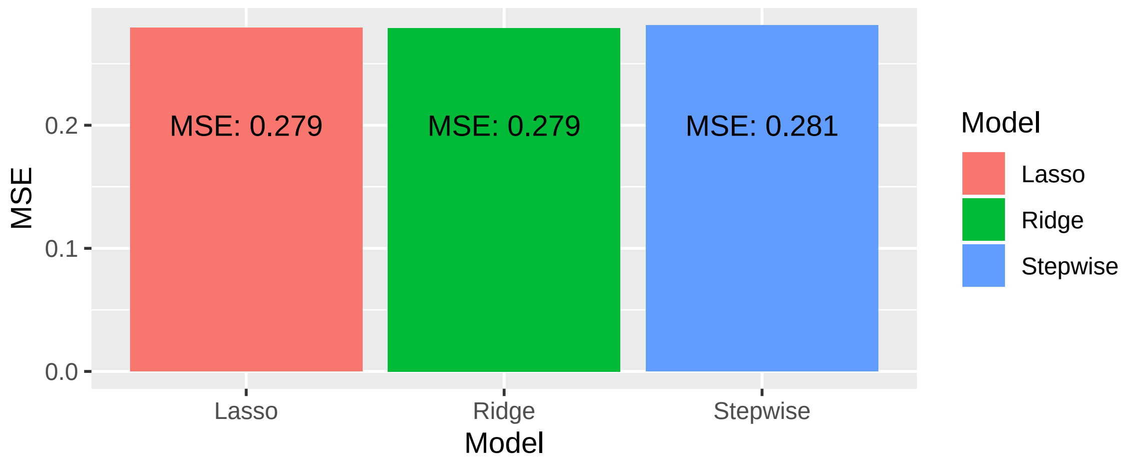


The L^2 norm for the coefficient vector is similar between each model, with the Ridge regression having the smallest value. This is not unexpected due to the penalization structure of Ridge regression. LASSO features a smaller L^1 norm than Ridge, but larger than the stepwise model. This is likely due to the five fewer predictors found in the stepwise model. Based on the plot and the results displayed in the table, these models all feature coefficient vectors of similar magnitude. In order to evaluate the effectiveness of each model, we plot our fitted and predicted values.









The plots of the actual vs. predicted values are nearly identical between models, as are the mean squared errors. The actual vs predicted plots suggest a non-normal error structure, which is to be expected as we have not transformed any of the variables or response. These figures allow us to draw the same conclusion as in problem 2, that the LASSO and ridge models are slightly better but by a seemingly insignificant amount. These methods are most useful for determining the most valuable predictors of the response, and in terms of the most effective and least effective predictors of `GFR`, the models were generally in agreement, which is what we would hope to find.

Biostats PhD Problems

1. Prove that the probability that an observation appears in a specific bootstrap sample is approximately 0.632. (Hint: Calculate the probability that the first (or any other) bootstrap observation is a specific observation (e.g., the jth) from the original dataset]

Let n be the number of observations of data that we are prepared to boostrap. Due to the nature of the with replacement independent sampling for bootstrap procedures, we have the following equivelencies:

$$\begin{aligned}
P(\text{obs } j \text{ is in boostrap }) &= 1 - P(\cap_{i=1}^{n}\{\text{obs } j \text{ not bootstrap selection } i\}) \\
&= 1 - \prod_{i=1}^{n} P(\{\text{obs } j \text{ not bootstrap selection } i\}) \\
&= 1 - \prod_{i=1}^{n}\left(1 - \frac{1}{n}\right) \\
&= 1 - \left(1 - \frac{1}{n}\right)^{n} \\
&= 1 - \left(1 + \frac{-1}{n}\right)^{n}
\end{aligned}$$

For large n, the second term in the final equality is approximately e^{-1} which means that

$$P(\text{obs } j \text{ is in boostrap }) \approx 1 - e^{-1} = 0.632$$

as desired.

2. Show that Ridge and LASSO minimization using a penalty terms is equivalent to the constrained optimization form. State the relationship between s and λ.

Consider the contrained optimization

$$\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \; s.t. \sum_{j=1}^{p}\beta_j^2 \leq s$$

We envoke Lagrange multipliers and define

$$\mathcal{L}(\beta_0, ..., \beta_p, \lambda) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 - \lambda s$$

For Ridge Regression, we are only concerned with detrimining the vector β that minimizes $\mathcal{L}$, and λ will be some fixed value, so the last term, λs vanishes when taking derivatives with respect to β. Similarly for LASSO, we can transform the constrained optimization

$$\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \; s.t. \sum_{j=1}^{p}|\beta_j| \leq s$$

into the Lagrangian

$$\mathcal{L}(\beta_0, ..., \beta_p, \lambda) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| - \lambda s$$

Again, since we are only minimizing this Lagrangian with respect to β, and the λs term is fixed, it does not influence the β vector that is the minimizer of $\mathcal{L}(\beta_0, ..., \beta_p, \lambda)$.

For equivalent optimization problems, in the case of both LASSO and Ridge, the relationship between λ and s is intuitively clear: as λ increases, s would decrease. This can be seen in the definition of $\mathcal{L}(\beta_0, ..., \beta_p, \lambda)$. For large λ, in order to minimize $\mathcal{L}(\beta_0, ..., \beta_p, \lambda)$, $\sum_{j=1}^{p} |\beta_j|$ in the case of LASSO and $\sum_{j=1}^{p} \beta_j^2$ in the case of Ridge are forced to be closer to zero, since λ is positively multiplying the L^1 and squared L^2 norm respectively. The 'forcing of the norm to be smaller' is exactly the purpose of s in the constrained optimization problem. For smaller s, the norm will be forced closer to zero.

3. Write the minimization problem in matrix form and then show that the ridge solution can be written as $\hat{\beta^{ridge}} = \left(X^T X + \lambda I\right)^{-1} X^T y$ where I is the identity matrix.

The Ridge Regresion minimization problem can be written

$$\hat{\beta}^{ridge} = \text{arg min}_{\beta} \left(y - X\beta\right)^T \left(y - X\beta\right) + \lambda \beta^t \beta$$

Differentiating the right side and setting equal to zero we obtain,

$$2X^T X\beta - 2X^T y + 2\lambda\beta = 0$$

Simplifying the above expression we have

$$\hat{\beta} = \left(X^T X + \lambda I\right)^{-1} X^T y$$

as desired.

4. Show that the ridge regression estimate is the mean (and mode) of the posterior distribution under a Gaussian prior $\beta \sim N(0, \tau^2 I)$ and Gaussian sampling model $y \sim N(X\beta, \sigma^2 I)$. Find the relationship between λ and the variances $\tau^2$and σ^2.

We have that

$$P(\beta|y) \propto P(y|\beta)P(\beta)$$

Given the distributional assumptions in the problem we can show

$$\begin{aligned} P(\beta|y) &\propto \exp\left\{ -\frac{\left(y - X\beta\right)^T \left(y - X\beta\right)}{2\sigma^2} \right\} \exp\left\{ -\frac{\beta^T \beta}{2\tau^2} \right\} \\ &= \exp\left\{ -\left(\frac{\left(y - X\beta\right)^T \left(y - X\beta\right)}{2\sigma^2} + \frac{\beta^T \beta}{2\tau^2} \right) \right\} \end{aligned}$$

Taking logs we have

$$\begin{aligned} \log(P(\beta|y)) &\propto -\log\left(\frac{\left(y - X\beta\right)^T \left(y - X\beta\right)}{2\sigma^2} + \frac{\beta^T \beta}{2\tau^2} \right) \\ &= -\log\left(\left(y - X\beta\right)^T \left(y - X\beta\right) + \frac{\sigma^2}{\tau^2} \beta^T \beta \right) \end{aligned}$$

We know tha the mean and mode are the β thar maximizes $\log(P(\beta|y))$. This is equivalent to minimizing $(y - X\beta)^T (y - X\beta) + \frac{\sigma^2}{\tau^2}\beta^T\beta$. Letting $\lambda = \sigma^2/\tau^2$ we have our equivalence to the Ridge Regression estimate.

5) Augment the centered design matrix X with p additional rows $\sqrt{\lambda}$ and augment y with p zeros. Show that the least squares solution to this augmented data set is thesame as the ridge solution to the original dataset. Therefore, by introducing artificialdata with response 0, the least squares coefficients are shrunk toward zero.

Let X be a $N \times p$ design matrix and y be an $n \times 1$ response vector. We define

$$Z = \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}, \quad w = \begin{bmatrix} y \\ \mathbf{0} \end{bmatrix}$$

so that Z is now $(N+p) \times p$ and w is $1 \times (N+p)$. The least squares solution involving design matrix Z and response w has the form

$$\begin{aligned} \hat{\beta} &= \left(Z^T Z\right)^{-1} Z^T w \\ &= \left(\begin{bmatrix} X^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} y \\ \mathbf{0} \end{bmatrix} \\ &= \left(X^T X + \lambda I\right)^{-1} X^T y \end{aligned}$$

Notice, this is just $\hat{\beta}^{ridge}$.

Code Appendix

```r
knitr::opts_chunk$set(echo = F)
knitr::opts_chunk$set(message = F)
knitr::opts_chunk$set(warning = F)
knitr::opts_chunk$set(fig.height = 3.5)
knitr::opts_chunk$set(fig.width = 6)
knitr::opts_chunk$set(fig.align="center")
`%notin%` <- Negate(`%in%`)
library(tidyverse)
library(caret)
library(leaps)

IOV = read.csv('iodatadev.csv')[,-1]

#Rename cols to match those in assignment
IOV = IOV %>% dplyr::rename(UPRO = upro, DIAB = Diabetes , CYS = cys,
                             HBPSTATUS = hbpstatus)

#select columns specified in assignment
ModelCols = c('WEIGHT', 'BMI', 'GFR', 'UCRE', 'UUN', 'UPHO', 'SUN','SCR',
              'TCHOL', 'ALB', 'HBA1C', 'PHOS', 'TRIG', 'LDL', 'HDL', 'HB',
              'MAP', 'UPRO', 'BSA', 'SODIUM', 'GLUC', 'BLACK', 'HEIGHT', 'AGE',
              'FEMALE', 'CYS', 'DBP', 'SBP', 'CRP', 'DIAB', 'HBPSTATUS')

IOV = IOV %>% dplyr::select(ModelCols)



library(visdat)

vis_miss(IOV,sort_miss = T) + theme(axis.text.x = element_text( size=4.5, angle = 60))

library(fBasics)
library(kableExtra)
#Check how many complete cases we have
#sum(complete.cases(IOV))/nrow(IOV)

#Look at particularly sparse columns
CheckNA = IOV[, c("UPHO", "HBA1C", "TRIG", "LDL", "HDL", "CRP", "HBPSTATUS")]

#See how many NAs they have
# summary(CheckNA)

NAsummary = round(basicStats(CheckNA)[c("Mean", "Stdev",  "Minimum", "1. Quartile", "Median", "3. Quart

NAsummary[8,] = as.character(as.integer(NAsummary[8,]))

rownames(NAsummary)[c(4,6)] = c("Q1", "Q3")

kable_styling(kable(NAsummary, escape = F, format = "latex", booktabs= TRUE, align = 'c'
                    ), latex_options = "HOLD_position")
```

```r
#Pick columns that arent in the sparse set
# GoodCols= which(colnames(IOV) %notin% c(colnames(CheckNA), "SBP"))
GoodCols = which(colnames(IOV) %notin% c(colnames(CheckNA)))

#sum(complete.cases(IOV[,GoodCols]))/nrow(IOV)

ReducedIOV = IOV[,GoodCols]
# sum(!complete.cases(ReducedIOV))
CompleteCases = which(complete.cases(ReducedIOV))
# print("Removed Cases")
# summary(ReducedIOV[-CompleteCases,c("UUN", "UCRE", "DIAB")])
# print("-----------------------------------------")
# print("Remaining Cases")
# summary(IOV[CompleteCases,c("UUN", "UCRE", "DIAB")])

RC = ReducedIOV[-CompleteCases,c("UUN", "UCRE", "DIAB")]
CC = ReducedIOV[CompleteCases,c("UUN", "UCRE", "DIAB")]

completeSummary = round(basicStats(CC)[c("Mean", "Stdev", "Median", "Minimum", "1. Quartile", "3. Quart

rownames(completeSummary)[c(5,6)] = c("Q1", "Q3")

completeSummary[8,] = as.character(as.integer(completeSummary[8,]))

kable_styling(kable(completeSummary, escape = F, format = "latex", booktabs= TRUE, align = 'l', caption
                    ), latex_options = "HOLD_position")

removedSummary = round(basicStats(RC)[c("Mean", "Stdev", "Median", "Minimum", "1. Quartile", "3. Quarti

rownames(removedSummary)[c(5,6)] = c("Q1", "Q3")
removedSummary[8,] = as.character(as.integer(removedSummary[8,]))

kable_styling(kable(removedSummary, escape = F, format = "latex", booktabs= TRUE, align = 'l', caption 
                    ), latex_options = "HOLD_position")

NewIOV = ReducedIOV[CompleteCases, -which(colnames(ReducedIOV)=="DIAB")]

#Fit Full Model

FullIOV = lm(GFR ~., data = NewIOV)
BackwardModel = step(FullIOV, trace = 0, direction = c('backward'))
# summary(BackwardModel)

MinIOV = lm(GFR ~1, data = NewIOV)
biggest = formula(BackwardModel)
ForwardModel = step(MinIOV, scope = biggest, trace = 0, direction = c('forward'))
# summary(ForwardModel)

# formula(ForwardModel)
# formula(BackwardModel)
set.seed(1234)
RsquareTest = function(test, model, data = NewIOV){
  testPreds = predict(model, newdata = data[test,])
  testResponse = data[test, ]$GFR

  SS.Total = sum((testResponse-mean(testResponse))^2)
  #SS.Regression = sum((testPreds - mean(testResponse))^2)
  SS.Residual = sum((testResponse - testPreds)^2)

  # rsq = SS.Regression/SS.Total
  rsq = 1 - SS.Residual/SS.Total
  return(rsq)
}

ForwardRsqTrain = rep(NA, 1000)
ForwardRsqTest = rep(NA,1000)
BackwardRsqTrain = rep(NA, 1000)
BackwardRsqTest = rep(NA,1000)
rowsIOV = 1:nrow(NewIOV)

for(i in 1:1000){
  sample_rows = sample(rowsIOV, replace = T)
  not_included = which(rowsIOV %notin% sample_rows)
  FullIOV = lm(GFR ~., data = NewIOV[sample_rows,])
  BackwardModel = step(FullIOV, trace = 0, direction = c('backward'))
  BackwardRsqTrain[i] = summary(BackwardModel)$r.squared
  BackwardRsqTest[i] = RsquareTest(not_included, BackwardModel)

  MinIOV = lm(GFR ~1, data = NewIOV[sample_rows,])
  biggest = formula(FullIOV)
  ForwardModel = step(MinIOV, scope = biggest, trace = 0, direction = c('forward'))
  ForwardRsqTrain[i] = summary(ForwardModel)$r.squared
  ForwardRsqTest[i] = RsquareTest(not_included, ForwardModel)
}

# mean(ForwardRsqTest)
# mean(BackwardRsqTest)
# mean(ForwardRsqTrain)
# mean(BackwardRsqTrain)
RsqBootOptBack = BackwardRsqTrain - BackwardRsqTest
MeanRsqBootOptBack = mean(RsqBootOptBack)

RsqBootOptForward = ForwardRsqTrain - ForwardRsqTest
MeanRsqBootOptForward = mean(RsqBootOptForward)
# BackwardTestREst = 0.721 - 0.017
# ForwardTestRest = 0.720 - 0.014
#
# mean(ForwardRsqTest)
# mean(BackwardRsqTest)
# mean(ForwardRsqTrain)
# mean(BackwardRsqTrain)

Prob1summary = data.frame(Model = c("Backwards", "Forwards"),
                          FullR2= c(0.721, 0.720),
                          BootR2Ave = c(0.725, 0.724),
                          BootTestR2Ave = c(0.708, 0.710),
                          Bootopt = c(0.017, 0.014),
                          EstTestR2 = c(0.704, 0.706)
                          )


library(kableExtra)
names(Prob1summary)[2] <- "$R^2_{full}$"
names(Prob1summary)[3] <- "Mean $R^2_{bootTrain}$"
names(Prob1summary)[4] <- "Mean $R^2_{bootTest}$"
names(Prob1summary)[5] <- "Mean $R^2_{bootOpt}$"
names(Prob1summary)[6] <- "Estimated Test $R^2$"
kable_styling(kable(Prob1summary, escape = F, format = "latex", booktabs= TRUE, align = 'c'
                    ))

ScaledNewIOV = data.frame(scale(NewIOV))

set.seed(1234)
Step_Train = sample(1:nrow(ScaledNewIOV), 0.75*nrow(ScaledNewIOV))

library(leaps)
bestmodels = regsubsets(GFR~., data = ScaledNewIOV[Step_Train,], nvmax = 22)
summ_bestmodels = summary(bestmodels)

get_model_formula <- function(id, object, outcome){
  models <- summary(object)$which[id,-1]
  predictors <- names(which(models == TRUE))
  predictors <- paste(predictors, collapse = "+")
  as.formula(paste0(outcome, "~", predictors))
}
# function taken from a reference page posted by Alboukadel Kassambara, PhD
# website http://www.sthda.com/

MSEs = rep(NA, 22)

for(i in 1:22){
  model = get_model_formula(i, bestmodels, "GFR")
  fit = lm(model, data = ScaledNewIOV[Step_Train,])
  y_resp = ScaledNewIOV[-Step_Train, "GFR"]
  y_preds = predict(fit, newdata = ScaledNewIOV[-Step_Train,])
  error = mean((y_resp - y_preds)^2)
  MSEs[i] = error
}

plot(1:22, MSEs, ylab= "MSE", xlab = "Number of Predictors", main = "Test MSE for Best Models with 'p' 
abline(h=min(MSEs), col = "red")



best_step_model = get_model_formula(9, bestmodels, "GFR")

refit_step = lm(best_step_model, data = ScaledNewIOV)

steprsq = summary(refit_step)$r.sq
```

```r
set.seed(1234)
library(glmnet)
lams = seq(0.01, 1, by = 0.01)
RidgeCV = cv.glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3], lambda = lams, alpl
                    nfolds = 5, type.measure = "mse")

best_lambda_Ridge = RidgeCV$lambda.min

ShrinkRidge = glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3], lambda = lams,
                    alpha = 0)

FinalRidge = glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3], lambda = best_lambd
                    alpha = 0)


par(mfrow= c(1,2))
plot(RidgeCV)
plot(ShrinkRidge, "lambda")
FinalRidge = glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3], lambda = best_lambd
                    alpha = 0)

RidgePreds = predict(FinalRidge, newx = model.matrix(GFR~., data = ScaledNewIOV)[,-1])[,1]
RidgeResps = ScaledNewIOV[,3]

SS.TotalR = sum((RidgeResps-mean(RidgeResps))^2)
SS.ResidualR = sum((RidgeResps - RidgePreds)^2)

RidgeRsquare = 1 - SS.ResidualR/SS.TotalR
set.seed(1234)
library(glmnet)
# lams = log(seq(exp(1), exp(0.005), length = 101))

lams = seq(0.001, 0.5, by = 0.001)

LASSOCV = cv.glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3], lambda = lams, alpl
                    nfolds = 5, type.measure = 'mse')

best_lambda_Lasso = LASSOCV$lambda.min

best_LASSO = glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3],
                    lambda = lams, alpha = 1)
par(mfrow = c(1,2))
plot(LASSOCV)
plot(best_LASSO, "lambda")
FinalLasso = glmnet(model.matrix(GFR~., data = ScaledNewIOV)[,-1], ScaledNewIOV[,3],
                    lambda = best_lambda_Lasso, alpha = 1)


LassoPreds = predict(FinalLasso, newx = model.matrix(GFR~., data = ScaledNewIOV)[,-1])
LassoResps = ScaledNewIOV[,3]

SS.TotalL = sum((LassoResps-mean(LassoResps))^2)
SS.ResidualL = sum((LassoResps - LassoPreds)^2)
```

```
LassoRsquare = 1 - SS.ResidualL/SS.TotalL
compareR2 = data.frame("Stepwise" = steprsq,
                       "Ridge" = RidgeRsquare,
                       "LASSO" = LassoRsquare,
                       "Ave Test $R^2$" = 0.705)

names(compareR2)[4] = "Ave Test $R^2$"
compareR2 = round(compareR2, 3)

kable_styling(kable(compareR2, escape = F, format = "latex", booktabs= TRUE, align = 'c'
                    ), latex_options = "HOLD_position")



# names(refit_step$coefficients)[-1]
#
# rownames(FinalRidge$beta)
#
# rownames(FinalLasso$beta)

modelComparison = data.frame("Stepwise" = rep(NA,23),
                             "Ridge" = rep(NA,23),
                             "LASSO"=  rep(NA,23))

rownames(modelComparison) = c("INTERCEPT", rownames(FinalRidge$beta))

modelComparison$Ridge = coefficients(FinalRidge)[,1]
modelComparison$LASSO[-1] = as.numeric(FinalLasso$beta)
modelComparison$LASSO[1] = coefficients(FinalLasso)[1]

for (i in 1:length(names(refit_step$coefficients)[-1])) {
  modelComparison[which(rownames(modelComparison) == names(refit_step$coefficients)[i+1]),1] = refit_st
}

modelComparison$Stepwise[1] = refit_step$coefficients[1]

modelComparison[which(is.na(modelComparison$Stepwise)),"Stepwise"] = 0

modelComparison=modelComparison[-1,]

# kable_styling(kable(round(modelComparison,4), escape = F, format = "latex", booktabs= TRUE, align = '
#                       caption = "Predictor Coefficients by Model"))


ggplot(data = ScaledNewIOV, aes(x=CYS, y=GFR)) + geom_point(color = "darkblue") +
  theme_bw() + ylab("Scaled GFR") + xlab("Scaled CYS")

ggplot(data = ScaledNewIOV, aes(x=SCR, y=GFR)) + geom_point(color = "darkred") +
  theme_bw() + ylab("Scaled GFR") + xlab("Scaled SCR")

# checkRidge = abs(modelComparison$Ridge)
# sort(checkRidge)
```

```
Norms = data.frame("Stepwise" = c(sqrt(sum(modelComparison$Stepwise^2)),
                                  sum(abs(modelComparison$Stepwise))),
                   "Ridge" = c(sqrt(sum(modelComparison$Ridge^2)),
                                  sum(abs(modelComparison$Ridge))),
                   "LASSO" = c(sqrt(sum(modelComparison$LASSO^2)),
                                  sum(abs(modelComparison$LASSO)))
)

rownames(Norms) = c("$L^2$", "$L^1$")
kable_styling(kable(round(Norms, 3), escape = F, format = "latex", booktabs= TRUE, align = 'c',
                     caption = "Coefficient Vector Norm Comparison"), latex_options = "HOLD_position")

Norms$Type = c("L2", "L1")

Norms_long <- gather(Norms, Model, Norm , -Type, factor_key=TRUE)


ggplot(data = Norms_long, aes(x = Type, y = Norm, group = Model )) + geom_bar(aes(fill = Model),stat =

StepPreds = predict(refit_step, newdata = ScaledNewIOV)
RidgePreds = predict(FinalRidge, newx = model.matrix(GFR~., data = ScaledNewIOV)[,-1])[,1]
LassoPreds = predict(FinalLasso, newx = model.matrix(GFR~., data = ScaledNewIOV)[,-1])[,1]

Predictions = data.frame(Response = ScaledNewIOV$GFR,
           RidgePreds = RidgePreds,
           StepPreds = StepPreds,
           LassoPreds = LassoPreds)


RidgeMSE = mean((RidgePreds - ScaledNewIOV$GFR)^2)
LassoMSE = mean((LassoPreds - ScaledNewIOV$GFR)^2)
StepMSE = mean((StepPreds - ScaledNewIOV$GFR)^2)



ModelMSE = data.frame(Model = c("Ridge", "Lasso", "Stepwise"),
           MSE = c(RidgeMSE, LassoMSE, StepMSE))





p1 = ggplot(Predictions, aes(x= RidgePreds, y = Response)) + geom_point() +
  geom_abline(slope = 1, color = "red", lty = 2) + ggtitle("Ridge Predicted vs. Actual")

p2 = ggplot(Predictions, aes(x= LassoPreds, y = Response)) + geom_point() +
  geom_abline(slope = 1, color = "red", lty = 2) + ggtitle("LASSO Predicted vs. Actual")

p3 = ggplot(Predictions, aes(x= StepPreds, y = Response)) + geom_point() +
  geom_abline(slope = 1, color = "red", lty = 2)+ ggtitle("Stepwise Predicted vs. Actual")

gridExtra::grid.arrange(p1,p2,p3)
ggplot(ModelMSE, aes(x = Model, y = MSE, fill = Model)) + geom_bar(stat = "identity") +
  annotate(geom = "text", x=1, y= 0.2, label = paste("MSE:",round(LassoMSE,3)))+
```

```r
  annotate(geom = "text", x=2, y= 0.2, label = paste("MSE:",round(RidgeMSE,3))) +
  annotate(geom = "text", x=3, y= 0.2, label = paste("MSE:",round(StepMSE,3)))
```