HW6: Model Selection & Multicollinearity (mostly ALRM Ch. 9, 10)

MATH/STAT 571A

DUE: 11/17/2023 11:59pm

Homework Guidelines

Please submit your answers on Gradescope as a PDF with pages matched to question answers.

One way to prepare your solutions to this homework is with R Markdown, which provides a way to include mathematical notation, text, code, and figures in a single document. A template `.Rmd` file is available through D2L.

Make sure all solutions are clearly labeled, and please utilize the question pairing tool on Gradescope. You are encouraged to work together, but your solutions, code, plots, and wording should always be your own. Come and see me and/or our TA during office hours or schedule an appointment when you get stuck and can't get unstuck.

I. Mathematical Foundations [14 pts]

The goal of the next two questions is to derive ALRM (10.21), $d_i = \frac{e_i}{1-h_{ii}}$, for the special case of simple linear regression through the origin (i.e., $Y_i = \beta X_i + \varepsilon_i$).

Let X and Y be the vectors $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_n)$, respectively, and let $X_{(i)}$ and $Y_{(i)}$ be the sub-vectors without the ith entry, such that $X_{(i)} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ and $Y_{(i)}$ defined analogously. Finally, let $b_{(i)} = (X'_{(i)}X_{(i)})^{-1}X'_{(i)}Y_{(i)}$ be the least-squares estimate of β based on $X_{(i)}$ and $Y_{(i)}$ and let $\hat{Y}_{i(i)} = X_i b_{(i)}$ denote the associated predicted value for the response at predictor level X_i.

(1) [3 pts] Show that for $h_{ii} = X_i(X'X)^{-1}X_i$, $X'_{(i)}X_{(i)} = X'X(1 - h_{ii})$ (*hint: note that* $X'X = X'_{(i)}X_{(i)} + X_i^2$).

(2) [5 pts] Use the result from (1) to show that $Y_i - \hat{Y}_{i(i)} = d_i = \frac{e_i}{1-h_{ii}}$.

The goal of the next question is to verify the formula for deleted residuals numerically for a model of spiny lobster density.

(3) [6 pts] Fit a linear regression model for the cube-root of lobster density using the predictors `"MPA"`, `"Depth_m"`, `"Relief_cm"`, `"Flat_Rock"`, `"Cobble"`, `"Boulder"`, `"Sand"` and the interaction between `"Depth_m"` and `"Flat_Rock"` (see question 5 from HW5), and verify the first residual value is 0.02743954. Next, fit the same model, but without the response or predictors for observation 1. Use the fitted model object to predict the cube-root of lobster density based on the predictor values associated with the 1st observation:

0.275 is the predicted.

```
predictors_rock <- c("MPA", "Depth_m", "Relief_cm",
                     "Flat_Rock", "Cobble", "Boulder", "Sand")
lobsters[1, predictors_rock]
```

```
##                             MPA Depth_m Relief_cm Flat_Rock Cobble Boulder Sand
## 1 Cabrillo State Marine Reserve     6.8      14.6     61.75    1.5   16.75   15
```

```
original_formula <- Lob_dens^(1/3) ~ MPA + Relief_cm +
                        Depth_m * Flat_Rock + Cobble + Boulder + Sand
lobsters_cubefit <- lm(original_formula, data = lobsters)
lobsters_cubefit$residuals[1] # matches
```

```
##          1
## 0.02743954
```

```
lobsters_cubefit2 <- lm(original_formula, data = lobsters[-1,])
pred_val <- predict(lobsters_cubefit2, newdata = lobsters[1, predictors_rock])
pred_val
```

```
##         1
## 0.2752032
```

```
# 0.275
real_val <- lobsters[1,"Lob_dens"]^(1/3)
real_val
```

```
## [1] 0.3072317
```

```
# 0.307 is the actual
```

Report the predicted value, $\hat{Y}_{1(1)}$**.** Confirm that the deleted residual obtained through re-fitting the model is the same as the one obtained using the formula $d_i = \frac{e_i}{1-h_{ii}}$.

```
X <- model.matrix(lobsters_cubefit)
H <- X %*% solve(t(X) %*% X) %*% t(X)
d_1 <- lobsters_cubefit$residuals[1]/(1-H[1,1])

c(real_val - pred_val, d_1)
```

```
##          1          1
## 0.03202846 0.03202846
```

II. Lobsters in Southern California [26 pts]

We will be using recently gathered data about the abundance of the California spiny lobster (*Panulirus interruptus*) along the southern coast of California. The California spiny lobster is an important commercial species for California, and there was concern that over-fishing might lead to a decline in the population of the species. In 2012, several marine protected areas (MPAs) were established where fishing is prohibited. The MPAs were not

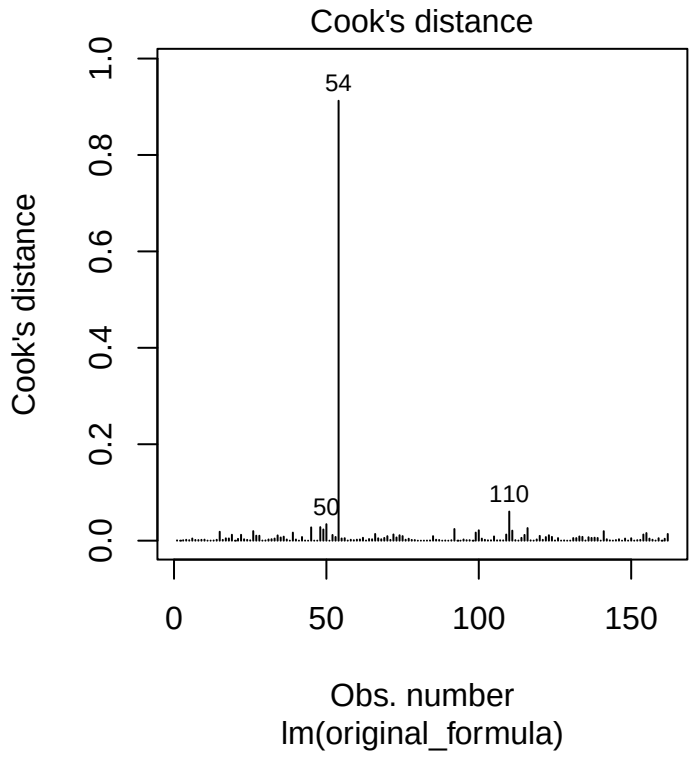
established explicitly to protect spiny lobsters, but they offer a natural experiment where abundance of spiny lobsters inside and outside the MPAs can be studied to see what effect the protected areas might have.

In 2012 and 2013, a team of researchers collected counts of spiny lobsters around 5 different MPAs along the coast near San Diego. Counts were made just inside and just outside each MPA. This **website** gives information about all the variables measured at each site, and this **website** gives the raw data (also posted on D2L). Your goal is to use linear models to study the relationships between the density of lobsters observed at each site and the characteristics of the sites.

(4) [3 pts] Make a plot of Cook's distances for each observation from the model fit in (3) with all observations. With which MPA is the observation with the largest Cook's distance associated? Is anything special/interesting about this observation? Do you see any reason why it should be removed from the analysis?

Observation 54 has MPA "Laguna Beach State Marine Reserve" associated with it. It has an abnormally tall relief structure (524 cm compared to 2nd tallest relief: 83.6 cm). Because the abnormally extreme relief height, lobster density may be affected differently by this one. Therefore, it seems reasonable to remove from this dataset.

```
plot(lobsters_cubefit, 4)
```




```
lobsters[54, predictors_rock]
```

```
##                                 MPA Depth_m Relief_cm Flat_Rock Cobble Boulder
## 54 Laguna Beach State Marine Reserve     7.3       524         0    7.5      77
##    Sand
## 54 15.5
```

```r
head(sort(lobsters$Relief_cm, decreasing = T))
```

```
## [1] 524.0  83.6  82.1  79.0  70.2  67.8
```

(5) [4 pts] Compute DFFITS for the model fit in (3). Which observation has the value with the largest magnitude? Compute DFBETAS for each regression coefficient. Which combination of observation and predictor variable has the value with the largest magnitude?

The observation with the largest magnitude DFFITS value is Obs. 54 with DFFITS: -3.309129. DFBETAS: Observation 54 with Relief_cm variable.

```r
# DFFITS
n <- nrow(lobsters)
p <- length(lobsters_cubefit$coefficients)
SSE <- sum(lobsters_cubefit$residuals^2)

e <- rep(0,n)
t <- rep(0,n)
DFFITS <- rep(0,n)

for (i in 1:n) {
  e[i] <- lobsters_cubefit$residuals[i]
  t[i] <- e[i] * ( (n-p-1) / ( SSE*(1-H[i,i]) - e[i]^2 ) )^(1/2)
  DFFITS[i] <- t[i] * ( (H[i,i]) / (1 - H[i,i]) )^(1/2)
}

largestDFFITS <- sort(DFFITS, decreasing = F)[1]
match(largestDFFITS,DFFITS)
```

```
## [1] 54
```

```r
# 54
lobsters[54, predictors_rock]
```

```
##                                  MPA Depth_m Relief_cm Flat_Rock Cobble Boulder
## 54 Laguna Beach State Marine Reserve     7.3       524         0    7.5      77
##    Sand
## 54 15.5
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# DFBETAS
DFBETAS <- as.data.frame(matrix(0, nrow=n, ncol=p, byrow=T))
parameters <- names(lobsters_cubefit$coefficients)
names(DFBETAS) <- parameters
rownames(DFBETAS) <- 1:n

for_c <- solve(t(X)%*%X)
for (i in 1:n){
  diff_fit <- lm(original_formula, data = lobsters[-c(i),])
  MSE_diff <- mean(diff_fit$residuals^2)

  for (k in 1:p) {
    b_k <- lobsters_cubefit$coefficients[[k]]
    b_k_diff <- diff_fit$coefficients[[k]]
    c_kk <- for_c[k,k]
    DFBETAS[i,k] <- (b_k - b_k_diff) / sqrt(MSE_diff * c_kk)
  }
}

big_beta <- max(-DFBETAS)
DFBETAS <- data.frame(DFBETAS, removed_obs = 1:n)
DFBETAS %>% filter(if_any(.cols = everything(),
                          .fns = ~ .x == -big_beta)
                   )
```

```
##   X.Intercept. MPALaguna.Beach.State.Marine.Reserve
## 1  -0.09398111                            0.1789852
##   MPAPoint.Vicente.State.Marine.Conservation.Area
## 1                                       0.3456741
##   MPASouth.La.Jolla.State.Marine.Reserve
## 1                             0.01795265
##   MPASwami.s.State.Marine.Conservation.Area Relief_cm   Depth_m Flat_Rock
## 1                              -0.007868252 -3.304546 -0.103675 0.1022428
##      Cobble   Boulder       Sand Depth_m.Flat_Rock removed_obs
## 1 0.1071444 0.1262625 0.09606784       0.009237251          54
```

```
# Obs 116 and Cobble
```

(6) [3 pts] Compute the sample correlations for all pairs of the variables "Depth_m", "Relief_cm", "Flat_Rock", "Cobble", "Boulder", "Sand". Do you see any evidence of multicollinearity? If so, which variables appear to be involved?

Yes, we see multicollinearity within two pairs of predictors: 1) Flat Rock and Boulder 2) Flat Rock and Sand

```
library(corrplot)
```

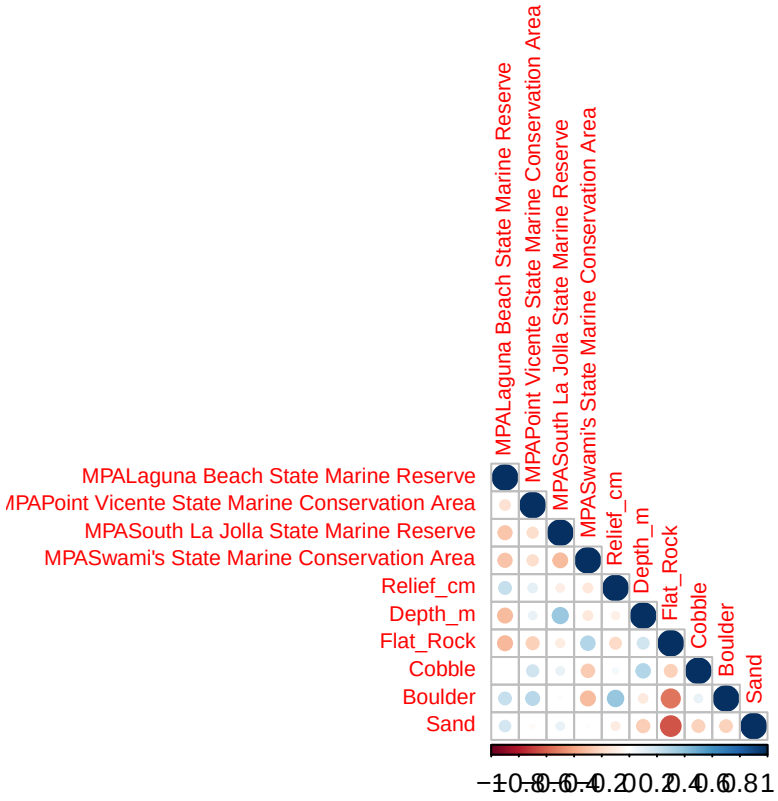
```
## Warning: package 'corrplot' was built under R version 4.2.3
```

```
## corrplot 0.92 loaded
```

```r
X <- model.matrix(lobsters_cubefit, data = lobsters)[,c(-1,-12)]
lobsters_cubefit$coefficients
```

```
##                                     (Intercept)
##                                   -1.453644e+00
##            MPALaguna Beach State Marine Reserve
##                                   -1.029688e-01
## MPAPoint Vicente State Marine Conservation Area
##                                   -2.643713e-01
##          MPASouth La Jolla State Marine Reserve
##                                    1.361404e-02
##       MPASwami's State Marine Conservation Area
##                                   -5.029377e-02
##                                       Relief_cm
##                                   -6.415588e-05
##                                         Depth_m
##                                   -4.293073e-03
##                                       Flat_Rock
##                                    2.051204e-02
##                                            Cobble
##                                    1.702630e-02
##                                         Boulder
##                                    1.988347e-02
##                                              Sand
##                                    1.682734e-02
##                                Depth_m:Flat_Rock
##                                   -2.716119e-04
```

```r
corrplot(cor(X), type = "lower", tl.cex = 0.65)
```



(7) [3 pts] Compute generalized variance inflation factors for all variables in the model (3). Do you see any evidence of multicollinearity? If so, which variables appear to be involved?

Flat_Rock and Depth_m appear to be involved.

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.2.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
vif(lobsters_cubefit, type = 'predictor')
```

```
## GVIFs computed for predictors
```

```
##                  GVIF Df GVIF^(1/(2*Df)) Interacts With
## MPA          2.347883  4        1.112588           --
## Relief_cm    1.233430  1        1.110599           --
## Depth_m   1362.577592  3        3.329611      Flat_Rock
## Flat_Rock 1362.577592  3        3.329611        Depth_m
## Cobble     112.086029  1       10.587069           --
## Boulder    401.788164  1       20.044654           --
## Sand       628.027424  1       25.060475           --
##                                            Other Predictors
## MPA       Relief_cm, Depth_m, Flat_Rock, Cobble, Boulder, Sand
## Relief_cm       MPA, Depth_m, Flat_Rock, Cobble, Boulder, Sand
## Depth_m                 MPA, Relief_cm, Cobble, Boulder, Sand
## Flat_Rock               MPA, Relief_cm, Cobble, Boulder, Sand
## Cobble       MPA, Relief_cm, Depth_m, Flat_Rock, Boulder, Sand
## Boulder       MPA, Relief_cm, Depth_m, Flat_Rock, Cobble, Sand
## Sand       MPA, Relief_cm, Depth_m, Flat_Rock, Cobble, Boulder
```

(8) [3 pts] Begin with the model from (3). Use the `step()` function in R to select a model using backwards step selection based on AIC. Which variables are removed compared the initial model?

Relief_cm was removed.

Backwards model: MPA, Depth_m, Flat_Rock, Cobble, Boulder, Sand, Depth_m:Flat_Rock

```
bw_step <- step(lobsters_cubefit, k = 2)
```

```
## Start:  AIC=-624.37
## Lob_dens^(1/3) ~ MPA + Relief_cm + Depth_m * Flat_Rock + Cobble +
##     Boulder + Sand
##
##                     Df Sum of Sq    RSS     AIC
## - Relief_cm          1   0.00101 2.9613 -626.32
## <none>                           2.9603 -624.37
## - Sand               1   0.05887 3.0192 -623.18
## - Cobble             1   0.05981 3.0202 -623.13
## - Boulder            1   0.08024 3.0406 -622.04
## - Depth_m:Flat_Rock  1   0.12500 3.0853 -619.67
## - MPA                4   0.83171 3.7920 -592.26
##
## Step:  AIC=-626.32
## Lob_dens^(1/3) ~ MPA + Depth_m + Flat_Rock + Cobble + Boulder +
##     Sand + Depth_m:Flat_Rock
##
##                     Df Sum of Sq    RSS     AIC
## <none>                           2.9613 -626.32
## - Sand               1   0.05791 3.0193 -625.18
## - Cobble             1   0.05884 3.0202 -625.13
## - Boulder            1   0.07924 3.0406 -624.04
## - Depth_m:Flat_Rock  1   0.12640 3.0878 -621.54
## - MPA                4   0.84722 3.8086 -593.56
```

```
original_var <- names(lobsters_cubefit$coefficients)
bw_var <- names(bw_step$coefficients)
```

```
original_var[!(original_var %in% bw_var)]
```

```
## [1] "Relief_cm"
```

(9) [3 pts] Use the `step()` function again to select a model, but this time start with a model that only includes the variables `"MPA"`, `"Depth_m"`, `"Relief_cm"` (still with a cube-root transformed response) and take steps in both directions. Were any variables included in the best model using backward step selection, but not included when taking steps in both directions? If so, which ones?

Step-wise in both direction led to less variables. The "Cobble" and "Sand" variables were eliminated. Backwards model: MPA, Depth_m, Flat_Rock, Cobble, Boulder, Sand, Depth_m:Flat_Rock Both model: MPA, Depth_m, Flat_Rock, Boulder, Depth_m:Flat_Rock

```
both_step <- step(lm(Lob_dens^(1/3) ~ MPA + Depth_m + Relief_cm, data = lobsters), k = 2,
     scope = list(upper = original_formula,
                  lower = Lob_dens^(1/3) ~ 1, direction = 'both'))
```

```
## Start:  AIC=-602.54
## Lob_dens^(1/3) ~ MPA + Depth_m + Relief_cm
##
##             Df Sum of Sq    RSS     AIC
## + Boulder    1   0.29947 3.3037 -614.59
## + Sand       1   0.28810 3.3151 -614.04
## - Relief_cm  1   0.03435 3.6375 -603.00
## <none>                   3.6032 -602.54
## + Flat_Rock  1   0.01760 3.5856 -601.33
## + Cobble     1   0.00005 3.6031 -600.54
## - Depth_m    1   0.28104 3.8842 -592.37
## - MPA        4   0.72470 4.3279 -580.85
##
## Step:  AIC=-614.59
## Lob_dens^(1/3) ~ MPA + Depth_m + Relief_cm + Boulder
##
##             Df Sum of Sq    RSS     AIC
## + Flat_Rock  1   0.14845 3.1552 -620.04
## + Sand       1   0.14323 3.1605 -619.77
## - Relief_cm  1   0.00000 3.3037 -616.59
## <none>                   3.3037 -614.59
## + Cobble     1   0.00001 3.3037 -612.59
## - Depth_m    1   0.20162 3.5053 -607.00
## - Boulder    1   0.29947 3.6032 -602.54
## - MPA        4   0.92530 4.2290 -582.59
##
## Step:  AIC=-620.04
## Lob_dens^(1/3) ~ MPA + Depth_m + Relief_cm + Boulder + Flat_Rock
##
##                     Df Sum of Sq    RSS     AIC
## + Depth_m:Flat_Rock  1   0.13507 3.0202 -625.13
## - Relief_cm          1   0.00047 3.1557 -622.02
## <none>                           3.1552 -620.04
## + Cobble             1   0.00994 3.1453 -618.55
## + Sand               1   0.00533 3.1499 -618.32
```

```
## - Flat_Rock          1   0.14845 3.3037 -614.59
## - Depth_m            1   0.27308 3.4283 -608.60
## - Boulder            1   0.43032 3.5856 -601.33
## - MPA                4   0.75211 3.9074 -593.41
##
## Step:  AIC=-625.13
## Lob_dens^(1/3) ~ MPA + Depth_m + Relief_cm + Boulder + Flat_Rock +
##     Depth_m:Flat_Rock
##
##                     Df Sum of Sq    RSS     AIC
## - Relief_cm          1   0.00004 3.0202 -627.13
## <none>                           3.0202 -625.13
## + Cobble             1   0.00097 3.0192 -623.18
## + Sand               1   0.00002 3.0202 -623.13
## - Depth_m:Flat_Rock  1   0.13507 3.1552 -620.04
## - Boulder            1   0.39873 3.4189 -607.04
## - MPA                4   0.81390 3.8341 -594.47
##
## Step:  AIC=-627.13
## Lob_dens^(1/3) ~ MPA + Depth_m + Boulder + Flat_Rock + Depth_m:Flat_Rock
##
##                     Df Sum of Sq    RSS     AIC
## <none>                           3.0202 -627.13
## + Cobble             1   0.00095 3.0193 -625.18
## + Relief_cm          1   0.00004 3.0202 -625.13
## + Sand               1   0.00002 3.0202 -625.13
## - Depth_m:Flat_Rock  1   0.13550 3.1557 -622.02
## - Boulder            1   0.44089 3.4611 -607.05
## - MPA                4   0.82643 3.8466 -595.94
```

```
both_var <- names(both_step$coefficients)

both_var
```

```
## [1] "(Intercept)"
## [2] "MPALaguna Beach State Marine Reserve"
## [3] "MPAPoint Vicente State Marine Conservation Area"
## [4] "MPASouth La Jolla State Marine Reserve"
## [5] "MPASwami's State Marine Conservation Area"
## [6] "Depth_m"
## [7] "Boulder"
## [8] "Flat_Rock"
## [9] "Depth_m:Flat_Rock"
```

```
bw_var
```

```
##  [1] "(Intercept)"
##  [2] "MPALaguna Beach State Marine Reserve"
##  [3] "MPAPoint Vicente State Marine Conservation Area"
##  [4] "MPASouth La Jolla State Marine Reserve"
##  [5] "MPASwami's State Marine Conservation Area"
##  [6] "Depth_m"
##  [7] "Flat_Rock"
```

```
##  [8] "Cobble"
##  [9] "Boulder"
## [10] "Sand"
## [11] "Depth_m:Flat_Rock"
```

```
both_var[!(both_var %in% bw_var)]
```

```
## character(0)
```

```
bw_var[!(bw_var %in% both_var)]
```

```
## [1] "Cobble" "Sand"
```

(10) [4 pts] Use the `glmnet` package to implement cross validation to estimate the mean-squared error across a range of values for the penalty term, λ, for a LASSO penalty. Determine and report a value for λ that seems optimal to you. Explain how you arrived at your value.

I chose 0.01684 as the lambda since it is 1 se above the minimum. Choosing this value will result in a more parsimonious model (less terms).

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.3
```

```
## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.2.3
```

```
## Loaded glmnet 4.1-7
```

```
X <- model.matrix(original_formula, data = lobsters)
cvfit <- cv.glmnet(x = X, y = lobsters$Lob_dens^(1/3), alpha = 1)
print(cvfit)
```

```
##
## Call:  cv.glmnet(x = X, y = lobsters$Lob_dens^(1/3), alpha = 1)
##
## Measure: Mean-Squared Error
##
##       Lambda Index Measure       SE Nonzero
## min 0.007289    23 0.02160 0.002687       6
## 1se 0.020282    12 0.02401 0.002925       5
```

(11) [3 pts] Fit the LASSO-penalized linear regression model with variables from (3) and your value of λ from the previous problem. Which variables have non-zero coefficients? How does your fit compare to the model selected using backwards stepwise selection in (8)?

LASSO removed Flat_Rock, Cobble, and Depth_m:Flat_Rock interaction term. LASSO model: MPA, Depth_m, Boulder, and Sand. Backwards model: MPA, Depth_m, Flat_Rock, Cobble, Boulder, Sand, Depth_m:Flat_Rock

```r
coef(glmnet(X, lobsters$Lob_dens^(1/3)), s = 0.02443)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                                                                 s1
## (Intercept)                                          0.2837468173
## (Intercept)                                          .
## MPALaguna Beach State Marine Reserve                 .
## MPAPoint Vicente State Marine Conservation Area -0.1197814750
## MPASouth La Jolla State Marine Reserve               .
## MPASwami's State Marine Conservation Area            .
## Relief_cm                                            .
## Depth_m                                             -0.0030278022
## Flat_Rock                                            .
## Cobble                                               .
## Boulder                                              0.0003744557
## Sand                                                -0.0004012408
## Depth_m:Flat_Rock                                    .
```