Homework 4 Peer Assessment Solutions

ISYE 6414 Instructor

Background

Selected molecular descriptors from the Dragon chemoinformatics application were used to predict bioconcentration factors for 779 chemicals in order to evaluate QSAR (Quantitative Structure Activity Relationship). This dataset was obtained from the UCI machine learning repository.

The dataset consists of 779 observations of 10 attributes. Below is a brief description of each feature and the response variable (logBCF) in our dataset:

1. *nHM* - number of heavy atoms (integer)
2. *piPC09* - molecular multiple path count (numeric)
3. *PCD* - difference between multiple path count and path count (numeric)
4. *X2Av* - average valence connectivity (numeric)
5. *MLOGP* - Moriguchi octanol-water partition coefficient (numeric)
6. *ON1V* - overall modified Zagreb index by valence vertex degrees (numeric)
7. *N.072* - Frequency of RCO-N< / >N-X=X fragments (integer)
8. *B02[C-N]* - Presence/Absence of C-N atom pairs (binary)
9. *F04[C-O]* - Frequency of C-O atom pairs (integer)
10. *logBCF* - Bioconcentration Factor in log units (numeric)

Note that all predictors with the exception of B02[C-N] are quantitative. For the purpose of this assignment, DO NOT CONVERT B02[C-N] to factor. Leave the data in its original format - numeric in R.

Please load the dataset "Bio_pred" and then split the dataset into a train and test set in a 80:20 ratio. Use the training set to build the models in Questions 1-6. Use the test set to help evaluate model performance in Question 7.R version 3.6.X and above, i.e. version 4.X is also acceptable.

Read Data

```r
set.seed(100)

fullData = read.csv("Bio_pred.csv",header=TRUE)
testRows = sample(nrow(fullData),0.2*nrow(fullData))
testData = fullData[testRows, ]
trainData = fullData[-testRows, ]

# Import the libraries
library(CombMSC)
library(boot)
library(leaps)
library(MASS)
library(glmnet)
```

Question 1: Full Model

(a) Fit a standard linear regression with the variable *logBCF* as the response and the other variables as predictors. Call it *model1*. Display the model summary.

```
model1 = lm(logBCF~.,data=trainData)
summary(model1)
```

```
##
## Call:
## lm(formula = logBCF ~ ., data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2577 -0.5180  0.0448  0.5117  4.0423
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.001422   0.138057   0.010  0.99179
## nHM          0.137022   0.022462   6.100 1.88e-09 ***
## piPC09       0.031158   0.020874   1.493  0.13603
## PCD          0.055655   0.063874   0.871  0.38391
## X2Av        -0.031890   0.253574  -0.126  0.89996
## MLOGP        0.506088   0.034211  14.793  < 2e-16 ***
## ON1V         0.140595   0.066810   2.104  0.03575 *
## N.072       -0.073334   0.070993  -1.033  0.30202
## B02.C.N.    -0.158231   0.080143  -1.974  0.04879 *
## F04.C.O.    -0.030763   0.009667  -3.182  0.00154 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7957 on 614 degrees of freedom
## Multiple R-squared:  0.6672, Adjusted R-squared:  0.6623
## F-statistic: 136.8 on 9 and 614 DF,  p-value: < 2.2e-16
```

Note: Using glm() instead of lm() is also acceptable.

(b) Which regression coefficients are significant at the 95% confidence level? At the 99% confidence level?

Regression coefficients that are statistically significant at the 99% confidence level:

```
names(which(summary(model1)$coefficients[,4]<.01))
```

```
## [1] "nHM"      "MLOGP"    "F04.C.O."
```

Regression coefficients that are statistically significant at the 95% confidence level:

```
names(which(summary(model1)$coefficients[,4]<.05))
```

```
## [1] "nHM"      "MLOGP"    "ON1V"     "B02.C.N." "F04.C.O."
```

nHM, MLOGP, and F04[C-O] are statistically significant at the 99% confidence level. nHM, MLOGP, ON1V, B02[C-N], and F04[C-O] are statistically significant at the 95% confidence level.

(c) What are the 10-fold and leave one out cross-validation scores for this model?

```
set.seed(100)

n=nrow(trainData)
modglm=glm(logBCF~.,data=trainData)
data.frame(ten.fold = cv.glm(trainData,modglm,K=10)$delta[1],
loocv = cv.glm(trainData,modglm,K=n)$delta[1])
```

```
##    ten.fold     loocv
## 1 0.6512928 0.6529872
```

The 10-fold CV score is 0.651292 while the leave one out score is 0.6529872.

Note: Using the caret package or any other alternate method is also acceptable. Depending on the function and cost function used (the default MSE cost is used here), the 10-fold CV score and LOOCV scores may be different.

(d) What are the Mallow's Cp, AIC, and BIC criterion values for this model?

```
set.seed(100)

# Show the Mallow Cp, AIC and BIC for this model
data.frame(Mallow = Cp(model1,S2=summary(model1)$sigma^2), AIC = AIC(model1,k=2),
BIC = AIC(model1,k=log(n)))
```

```
##   Mallow      AIC      BIC
## 1     10 1497.477 1546.274
```

(e) Build a new model on the training data with only the variables which coefficients were found to be statistically significant at the 99% confident level. Call it *model2*. Perform an ANOVA test to compare this new model with the full model. Which one would you prefer? Is it good practice to select variables based on statistical significance of individual coefficients? Explain.

```
set.seed(100)
# Create a new model with only the statistically significant predictors
model2 = lm(logBCF ~ nHM + MLOGP + F04.C.O., data = trainData)
summary(model2)
```

```
##
## Call:
## lm(formula = logBCF ~ nHM + MLOGP + F04.C.O., data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2555 -0.5097  0.0374  0.5471  4.2704
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03076    0.07836  -0.393   0.6948
## nHM          0.10948    0.01762   6.213 9.56e-10 ***
## MLOGP        0.60993    0.02177  28.018  < 2e-16 ***
## F04.C.O.    -0.01295    0.00745  -1.738   0.0826 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8037 on 620 degrees of freedom
## Multiple R-squared:  0.6571, Adjusted R-squared:  0.6554
## F-statistic:   396 on 3 and 620 DF,  p-value: < 2.2e-16
```

```
# Run ANOVA comparing the new model and the full model
anova(model2, model1)
```

```
## Analysis of Variance Table
##
## Model 1: logBCF ~ nHM + MLOGP + F04.C.O.
## Model 2: logBCF ~ nHM + piPC09 + PCD + X2Av + MLOGP + ON1V + N.072 + B02.C.N. +
##     F04.C.O.
##   Res.Df    RSS Df Sum of Sq     F  Pr(>F)
## 1    620 400.51
## 2    614 388.70  6    11.809 3.109 0.00523 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A p-value of 0.00523 suggests that we reject the null hypothesis that all the additional predictor coefficients in the full model are zero. At least one of the predictors excluded in the reduced model has a coefficient not equal to zero and hence, the full model is preferred over the reduced model.

This approach is generally not recommended for variable selection. It is NOT good practice to select variables based on statistical significance of individual coefficients.See Lesson 5.5.2.

Question 2: Full Model Search

(a) Compare all possible models using Mallow's Cp. What is the total number of possible models with the full set of variables? Display a table indicating the variables included in the best model of each size and the corresponding Mallow's Cp value.

Hint: You can use nbest parameter.

```
set.seed(100)
col_names = names(trainData)[-10]
out = leaps(trainData[,-10], trainData$logBCF, method = "Cp", nbest=1, names = col_names)
cbind(as.matrix(out$which),out$Cp)
```

```
##   nHM piPC09 PCD X2Av MLOGP ON1V N.072 B02.C.N. F04.C.O.
## 1   0      0   0    0     1    0     0        0        0 58.596851
## 2   1      0   0    0     1    0     0        0        0 17.737801
## 3   1      1   0    0     1    0     0        0        0 15.184626
## 4   1      1   0    0     1    0     0        0        1  9.495041
## 5   1      1   0    0     1    0     0        1        1  7.240754
## 6   1       1   0    0     1    1     0         1         1  6.116174
## 7   1       1   0    0     1    1     1         1         1  6.831852
## 8   1       1   1    0     1    1     1         1         1  8.015816
## 9   1       1   1    1     1    1     1         1         1 10.000000
```

There are $2^9 = 512$ possible models, since there are 9 predictors.

(b) How many variables are in the model with the lowest Mallow's Cp value? Which variables are they? Fit this model and call it *model3*.

```
set.seed(100)
# Get the index of the best model
best.model = which(out$Cp==min(out$Cp))
# Extract the coefficients of the best model
best.cp = cbind(as.matrix(out$which),out$Cp)[best.model,]
# Set the names of the best row
names(best.cp) = c(names(trainData[,-c(10)]),"Cp")
# Show the best row with the selected predictors
best.cp
```

```
##      nHM   piPC09      PCD     X2Av    MLOGP     ON1V    N.072 B02.C.N.
## 1.000000 1.000000 0.000000 0.000000 1.000000 1.000000 0.000000 1.000000
## F04.C.O.       Cp
## 1.000000 6.116174
```

```
# Create the model using the selected predictors
model3 = lm(formula = logBCF ~ . - PCD - X2Av - N.072, data = trainData)
summary(model3)
```

```
##
## Call:
## lm(formula = logBCF ~ . - PCD - X2Av - N.072, data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2364 -0.5234  0.0421  0.5196  4.1159
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.035785   0.099454   0.360  0.71911
## nHM          0.124086   0.019083   6.502 1.63e-10 ***
## piPC09       0.042167   0.014135   2.983  0.00297 **
## MLOGP        0.528522   0.029434  17.956  < 2e-16 ***
## ON1V         0.098099   0.055457   1.769  0.07740 .
## B02.C.N.    -0.160204   0.073225  -2.188  0.02906 *
## F04.C.O.    -0.028644   0.009415  -3.042  0.00245 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7951 on 617 degrees of freedom
## Multiple R-squared:  0.666,  Adjusted R-squared:  0.6628
## F-statistic: 205.1 on 6 and 617 DF,  p-value: < 2.2e-16
```

There are 6 variables in the model with the lowest Mallow's CP value. The variables are nHM, piPC09, MLOGP, ON1V, B02[C-N], and F04[C-O].

Question 3: Stepwise Regression

(a) Perform backward stepwise regression using BIC. Allow the minimum model to be the model with only an intercept, and the full model to be *model1*. Display the model summary of your final model. Call it *model4*

```r
set.seed(100)
# Create the minimum model
minmod = lm(logBCF~1, data = trainData)
# Step backward from the full model using BIC as the complexity penalty
model4 = step(model1, scope = list(lower = minmod, upper = model1),
              direction = "backward", k=log(n), trace=F)
# Show the optimal model
summary(model4)
```

```
##
## Call:
## lm(formula = logBCF ~ nHM + piPC09 + MLOGP + F04.C.O., data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2611 -0.5126  0.0517  0.5353  4.3488
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.008695   0.078196  -0.111  0.91150
## nHM          0.114029   0.017574   6.489 1.78e-10 ***
## piPC09       0.041119   0.013636   3.015  0.00267 **
## MLOGP        0.566473   0.025990  21.796  < 2e-16 ***
## F04.C.O.    -0.022104   0.008000  -2.763  0.00590 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7985 on 619 degrees of freedom
## Multiple R-squared:  0.662,  Adjusted R-squared:  0.6599
## F-statistic: 303.1 on 4 and 619 DF,  p-value: < 2.2e-16
```

Note: Using glm instead of lm is also acceptable.

(b) How many variables are in *model4*? Which regression coefficients are significant at the 99% confidence level?

Regression coefficients that are statistically significant at the 99% confidence level:

```r
names(which(summary(model4)$coefficients[,4]<.01))
```

```
## [1] "nHM"      "piPC09"   "MLOGP"    "F04.C.O."
```

There are 4 variables in model 4. All regression coefficients except the intercept are statistically significant in this model, which include nHM, piPC09, MLOGP, and F04[C-O].

(c) Perform forward stepwise selection with AIC. Allow the minimum model to be the model with only an intercept, and the full model to be *model1*. Display the model summary of your final model. Call it *model5*. Do the variables included in *model5* differ from the variables in *model4*?

```r
set.seed(100)
# Step forward from the minimum model using AIC as the complexity penalty
model5 = step(minmod, scope = list(lower = minmod, upper = model1),
direction = "forward", k=2, trace=F)
summary(model5)
```

```
##
## Call:
## lm(formula = logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N. +
##     ON1V, data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2364 -0.5234  0.0421  0.5196  4.1159
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.035785   0.099454   0.360  0.71911
## MLOGP        0.528522   0.029434  17.956  < 2e-16 ***
## nHM          0.124086   0.019083   6.502 1.63e-10 ***
## piPC09       0.042167   0.014135   2.983  0.00297 **
## F04.C.O.    -0.028644   0.009415  -3.042  0.00245 **
## B02.C.N.    -0.160204   0.073225  -2.188  0.02906 *
## ON1V         0.098099   0.055457   1.769  0.07740 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7951 on 617 degrees of freedom
## Multiple R-squared:  0.666,  Adjusted R-squared:  0.6628
## F-statistic: 205.1 on 6 and 617 DF,  p-value: < 2.2e-16
```

Yes, the variables in model5 differ from the variables in model4. Model4 has 4 variables, while model5 has 6 variables. Model5 contains all the variables in model4 plus B02[C-N] and ON1V.

(d) Compare the adjusted R^2, Mallow's Cp, AICs and BICs of the full model(*model1*), the model found in Question 2 (*model3*), and the model found using backward selection with BIC (*model4*). Which model is preferred based on these criteria and why?

```r
set.seed(100)

comp = rbind(Q1.Full=c(summary(model1)$adj.r.sq,
                       Cp(model1,S2=summary(model1)$sigma^2),
                       AIC(model1,k=2), AIC(model1,k=log(n))),
             Q2.Complete.Search=c(summary(model3)$adj.r.sq,
                       Cp(model3,S2=summary(model1)$sigma^2),
                       AIC(model3,k=2), AIC(model3,k=log(n))),
             Q3.Backward.BIC=c(summary(model4)$adj.r.sq,
                       Cp(model4,S2=summary(model1)$sigma^2),
                       AIC(model4,k=2), AIC(model4,k=log(n))))
colnames(comp) = c("adj.rsq","Cp","AIC", "BIC")
comp
```

```
##                        adj.rsq        Cp      AIC      BIC
## Q1.Full              0.6623027 10.000000 1497.477 1546.274
## Q2.Complete.Search 0.6627864  6.116174 1493.623 1529.113
## Q3.Backward.BIC    0.6598504  9.495041 1497.052 1523.669
```

Model3 is generally preferred on these criteria as the model constructed from Mallow's CP exhaustive search scores the best out of all three models in three out of the four categories (adjusted R^2, Mallow's CP, and AIC). However, it can also be argued that Model4 which scores the best on BIC would be favored in a setting to reduce complexity and avoid overfitting.

Note: Any other reasonable explanation is also acceptable.

Question 4: Ridge Regression

(a) Perform ridge regression on the training set. Use cv.glmnet() to find the lambda value that minimizes the cross-validation error using 10 fold CV.

Note: By default, the glmnet() function standardizes the variables so that they are on the same scale. If a student implemented the standardization separetely and it appears correct and reasonable, please award full points.

```
set.seed(100)
# Optimize lambda using cross validation
cv.ridge = cv.glmnet(as.matrix(trainData[,-10]), trainData[,10],
                     family='gaussian', alpha=0, nfolds=10)
cat("CV Optimized lambda:\n")
```

```
## CV Optimized lambda:
```

```
cv.ridge$lambda.min
```

```
## [1] 0.108775
```

The lambda value that minimizes 10-fold CV error is 0.108775.

(b) List the value of coefficients at the optimum lambda value.

```
set.seed(100)
ridge.mod = glmnet(as.matrix(trainData[,-10]), trainData[,10],
                     family='gaussian', alpha=0)
# Extract coefficents at optimal lambda
coef(ridge.mod, s = cv.ridge$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.13841426
## nHM          0.14391877
## piPC09       0.03735762
## PCD          0.08235334
## X2Av        -0.06901352
## MLOGP        0.44403654
## ON1V         0.15770114
## N.072       -0.09683534
## B02.C.N.    -0.20919397
## F04.C.O.    -0.03177144
```

(c) How many variables were selected? Give an explanation for this number.

All 9 variables were selected. Ridge regression does not perform variable selection, only shrinkage. This is because it uses the L2 norm for penalization, which does not measure sparsity.

Question 5: Lasso Regression

(a) Perform lasso regression on the training set. Use cv.glmnet() to find the lambda value that minimizes the cross-validation error using 10 fold CV.

```r
set.seed(100)

# Optimize lambda using cross validation
cv.lasso = cv.glmnet(as.matrix(trainData[,-10]), trainData[,10],
                     family='gaussian', alpha=1, nfolds=10)
cat("CV Optimized lambda:\n")
```

```
## CV Optimized lambda:
```

```r
cv.lasso$lambda.min
```

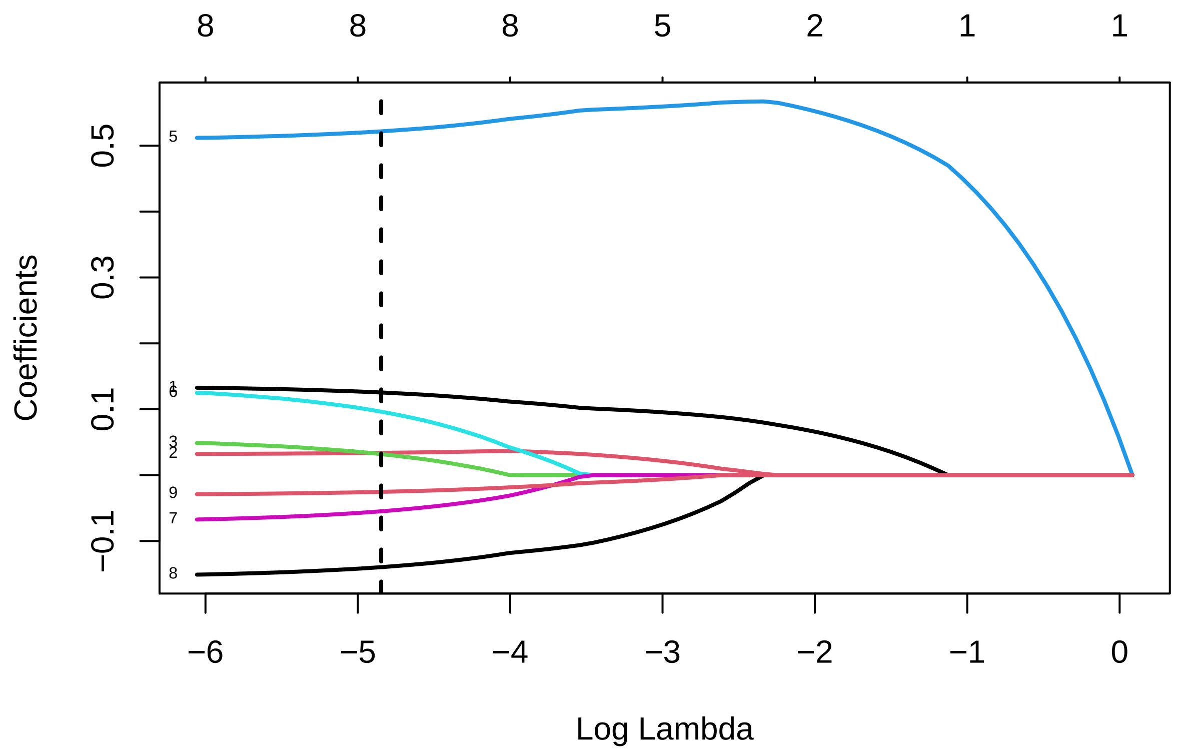
```
## [1] 0.007854436
```

The lambda value that minimizes the 10-fold CV error is 0.007854436.

(b) Plot the regression coefficient path.

```r
set.seed(100)

model6 = glmnet(as.matrix(trainData[,-10]), trainData[,10], family='gaussian', alpha=1)
plot(model6,xvar="lambda",label=TRUE,lwd=2)
abline(v=log(cv.lasso$lambda.min),col='black',lty = 2,lwd=2)
```



(c) How many variables were selected? Which are they?

```
set.seed(100)

# Extract coefficents at optimal lambda
coef(model6,s=cv.lasso$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.02722838
## nHM          0.12543866
## piPC09       0.03387665
## PCD          0.03194878
## X2Av         .
## MLOGP        0.52174346
## ON1V         0.09633951
## N.072       -0.05487196
## B02.C.N.    -0.13961811
## F04.C.O.    -0.02535576
```

At the optimal lambda, 8 variables were selected. This includes all variables except X2Av (nHM, piPC09, PCD, MLOGP, ON1V, N.072, B02[C-N], F04[C-O]).

Question 6: Elastic Net

(a) Perform elastic net regression on the training set. Use cv.glmnet() to find the lambda value that minimizes the cross-validation error using 10 fold CV. Give equal weight to both penalties.

```
set.seed(100)

# Optimize lambda using cross validation
cv.elnet=cv.glmnet(as.matrix(trainData[,-10]), trainData[,10],
                   family='gaussian', alpha = 0.5, nfolds=10)
model7 = glmnet(as.matrix(trainData[,-10]), trainData[,10],
                family='gaussian', alpha = 0.5)
cv.elnet$lambda.min
```

```
## [1] 0.0207662
```

The lambda value that minimizes elastic net 10-fold CV error is 0.0207662.

(b) List the coefficient values at the optimal lambda. How many variables were selected? How do these variables compare to those from Lasso in Question 5?

```
set.seed(100)

# Extract coefficents at optimal lambda
coef(model7,s=cv.elnet$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.04903516
## nHM          0.12397290
## piPC09       0.03470891
## PCD          0.03060034
## X2Av         .
## MLOGP        0.51776470
## ON1V         0.08901088
## N.072       -0.05236840
## B02.C.N.    -0.14155538
## F04.C.O.    -0.02420217
```

Coefficients are shown in the output above. Similar to Lasso, 8 out of 9 variables are selected. Lasso and Elastic Net chose the same set of variables.

Question 7: Model comparison

(a) Predict *logBCF* for each of the rows in the test data using the full model, and the models found using backward stepwise regression with BIC, ridge regression, lasso regression, and elastic net.

Note: A submission could have used either Lasso or a retrained Lasso model. The solutions below show both.

Note: If a submission standardized the trainData manually, then it must also standardize the testData before making predictions.

```r
set.seed(100)

# Full model
full = predict(model1,testData)
# Backward SR
bstepwise = predict(model4,testData)
# Ridge
ridge = as.vector(predict(ridge.mod, as.matrix(testData[,-10]), s=cv.ridge$lambda.min))
# Lasso
lasso = as.vector(predict(model6, as.matrix(testData[,-10]), s=cv.lasso$lambda.min))
# Retrained OLS model using Lasso-selected predictors
model6.retrained = lm(logBCF~ .-X2Av, data = trainData)
lasso.retrained = as.vector(predict(model6.retrained, testData))
# Elastic Net
elastic = as.vector(predict(model7,as.matrix(testData[,-10]), s=cv.elnet$lambda.min))

# Create a data frame with the predictions
preds = data.frame(logBCF = testData$logBCF, full, bstepwise,
                   ridge, lasso, lasso.retrained, elastic)
head(preds,3)
```

```
##     logBCF     full bstepwise    ridge    lasso lasso.retrained  elastic
## 714   2.64 2.446479  2.424916 2.454878 2.442895        2.446392 2.441506
## 503   4.58 4.333759  4.353167 4.234425 4.313509        4.333415 4.296451
## 358   3.44 3.266892  3.274192 3.223166 3.260617        3.267254 3.252638
```

Predictions for each test row along with the true value are shown above.

(b) Compare the predictions using mean squared prediction error. Which model performed the best?

```r
set.seed(100)

sapply(preds[,-1], function(x){mean((x-testData$logBCF)^2)})
```

```
##            full       bstepwise           ridge           lasso lasso.retrained 
##       0.5839296       0.5742198       0.5877835       0.5790832       0.5835470 
##         elastic 
##       0.5782750
```

The model that performed the best on the test set (using mean squared prediction error) was the backwards stepwise model. Its mspe value of 0.5742198 was slightly lower than the next closest model - elastic net at 0.5782750 mspe.

Note: Any other reasonable explanation is also acceptable.

(c) Provide a table listing each method described in Question 7a and the variables selected by each method (see Unit 5.2.3 for an example). Which variables were selected consistently?

	Backward Stepwise	Ridge	Lasso	Elastic Net
nHM	X	X	X	X
piPC09	X	X	X	X
PCD		X	X	X
X2AV		X		
MLOGP	X	X	X	X
ON1V		X	X	X
N.072		X	X	X
B02.C.N.		X	X	X
F04.C.O.	X	X	X	X

The variables nHM, piPC09, MLOGP, and F04[C-O] were selected consistently across all models. Note that while the Backwards Stepwise model eliminated PCD, ON1V, N.072, and B02[C-N], Lasso and elastic net did not. X2AV was eliminated from all the models that performed subset selection.

Note: The table can be formatted differently and made in any way. What's important is that it's easy to read and understand and that the correct variables are selected given their responses to the previous questions.