Homework 5 Peer Assessment

Fall Semester 2021

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. Please make sure that you are using R version 3.6.X or above (i.e. version 4.X is also acceptable).

Read Data

```
# Clear variables in memory
rm(list=ls())

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

# Ensure that the sampling type is correct
RNGkind(sample.kind="Rejection")

# Set a seed for reproducibility
```

```r
set.seed(100)

# Read data
fullData = read.csv("Bio_pred.csv",header=TRUE)

# Split data for traIning and testing
testRows = sample(nrow(fullData),0.2*nrow(fullData))
testData = fullData[testRows, ]
trainData = fullData[-testRows, ]
```

Note: Use the training set to build the models in Questions 1-6. Use the test set to help evaluate model performance in Question 7.

Question 1: Full Model

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

```r
#Fitting MLR Model - Full Model
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
```

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

Statistical Significance at 99% Confidence Level (α=0.01):
nHM, MLOGP, F04.C.O.

Statistical Significance at 95% Confidence Level (α=0.05):
nHM, MLOGP, ON1V, B02.C.N., F04.C.O.

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

```
#Calculating Mallow's CP, AIC, BIC
set.seed(100)
n = nrow(trainData)

c(Cp(model1, S2=summary(model1)$sigma^2),AIC(model1, k=2),AIC(model1,k=log(n)))
```

```
## [1]   10.000 1497.477 1546.274
```

Mallow's CP: 10.000
AIC: 1497.477
BIC: 1546.274

(d) 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 a Partial F-test to compare this new model with the full model (*model1*). Which one would you prefer? Is it good practice to select variables based on statistical significance of individual coefficients? Explain.

```
#Fitting MLR - Reduced Model (99% Conf Lvl)
set.seed(100)
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
```

```r
#Partial F-Test - ANOVA
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
```

H_0: α_1=α_2=...=0 (coefficients associated with piPC09, PCD, X2Av, ON1V, N.072, B02.C.N.)
H_A: at least one of the α coefficients is not equal to zero

The F-value is equal to 3.109 with a corresponding p-value of 0.00523. Given that the f-value is sufficiently large and the p-value is sufficiently small, we reject the null hypothesis and conclude that at least one of the additional predictors in the full model (piPC09, PCD, X2Av, ON1V, N.072, B02.C.N.) will be significantly associated with the response variable, logBCF.

As a result, the full model is preferred over the reduced model. In general, it is not recommended to select variables based on statistical significance. A regression coefficient may not be statistically significant in the full model, but once another predicting variable is discarded, it may then become statistically significant and vice versa.

Question 2: Full Model Search

(a) Compare all possible models using Mallow's Cp. How many models can be constructed using subsets/combinations drawn from 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.

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

From the full set of predicting variables in the original data set, we can construct a total of 512 (2^9) models using different subsets/combinations of variables.

Models with the number of variables ranging from 1 to 9 are displayed with their corresponding Mallow's CP.

(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*. Display the model summary.

```
#Determine "Best" Model - Mallow's CP
set.seed(100)

out=leaps(trainData[,-c(10)], trainData$logBCF, method="Cp", names = col_names)
best.model=which(out$Cp==min(out$Cp))
cbind(as.matrix(out$which),out$Cp)[best.model,]
```

```
##      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.
## 1.000000 6.116174
```

The best model with respect to the Mallow's CP includes 6 variables: nHM, piPC09, MLOGP, ON1V, B02.C.N., F.04.C.0. This model has a corresponding Mallow's CP of 6.116174.

```
#Fitting MLR - Best Mallow's CP
model3 = lm(logBCF ~ nHM + piPC09 + MLOGP + ON1V + B02.C.N. + F04.C.O., data=trainData)
summary(model3)
```

```
##
## Call:
## lm(formula = logBCF ~ nHM + piPC09 + MLOGP + ON1V + B02.C.N. +
##     F04.C.O., 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
```

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*

```
#Backward Stepwise Regression
set.seed(100)

min = lm(logBCF ~ 1, data=trainData)
model4 = step(model1, scope=list(lower=min, upper=model1), direction="backward", k=log(n), trace=F)
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
```

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

Model 4 has a total of 4 variables, plus an intercept. All four variables are statistically significant at an α level of 0.01: nHM, piPC09, MLOGP, 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*?

```
#Forward Stepwise Regression
set.seed(100)

model5 = step(min,scope=list(lower=min, 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
```

Comparing Models 4 and 5, Model 4 contains 4 variables plus an intercept while Model 5 has 6 variables plus an intercept. Model 5 includes all the same variables as model 4 with the addition of B.02.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?

```
c(Cp(model1, S2=summary(model1)$sigma^2),AIC(model1, k=2),AIC(model1,k=log(n)))
```

```
## [1]   10.000 1497.477 1546.274
```

```
c(Cp(model3, S2=summary(model1)$sigma^2),AIC(model3, k=2),AIC(model3,k=log(n)))
```

```
## [1]    6.116174 1493.623474 1529.112677
```

```
c(Cp(model4, S2=summary(model1)$sigma^2),AIC(model4, k=2),AIC(model4,k=log(n)))
```

```
## [1]    9.495041 1497.052364 1523.669266
```

Model	Adjusted R^2	Mallow's Cp	AIC	BIC
Model1	0.6623	10.00000	1497.477	1546.274
Model3	0.6628	6.116174	1493.623	1529.113
Model4	0.6599	9.495041	1497.052	1523.669

From the comparison across all four indicators, Model 3 is the preferred model. All models are fairly even with respect to the adjusted R^2 value, AIC, and BIC. The main differentiating factor is the Mallow's CP where Model 3 has the lowest value.

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.

```
#Optimal Lambda Value - Ridge
set.seed(100)
r.cv = cv.glmnet(as.matrix(trainData[,-10]), trainData[,10], family="gaussian", alpha=0, nfolds=10)

r.cv$lambda.min
```

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

Lambda value that minimizes 10-fold CV error = 0.108775

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

```
#Ridge Coefficients @ Lambda
set.seed(100)
model.ridge = glmnet(as.matrix(trainData[,-10]), trainData[,10], family="gaussian", alpha=0, nfolds=10)

coef(model.ridge, s = r.cv$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.13841426
## nHM          0.14391877
## piPC09       0.03735762
## PCD          0.08235334
## X2Av        -0.06901352
## MLOGP        0.44403655
## ON1V         0.15770114
## N.072       -0.09683534
## B02.C.N.    -0.20919397
## F04.C.O.    -0.03177144
```

(c) How many variables were selected? Was this result expected? Explain.

All the original 9 variables were selected. This is an expected result since ridge regression does not perform variable selection, only shrinkage.

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.

```
set.seed(100)
#Optimal Lambda - Lasso
l.cv = cv.glmnet(as.matrix(trainData[,-10]), trainData[,10], family="gaussian", alpha=1, nfolds=10)

l.cv$lambda.min
```

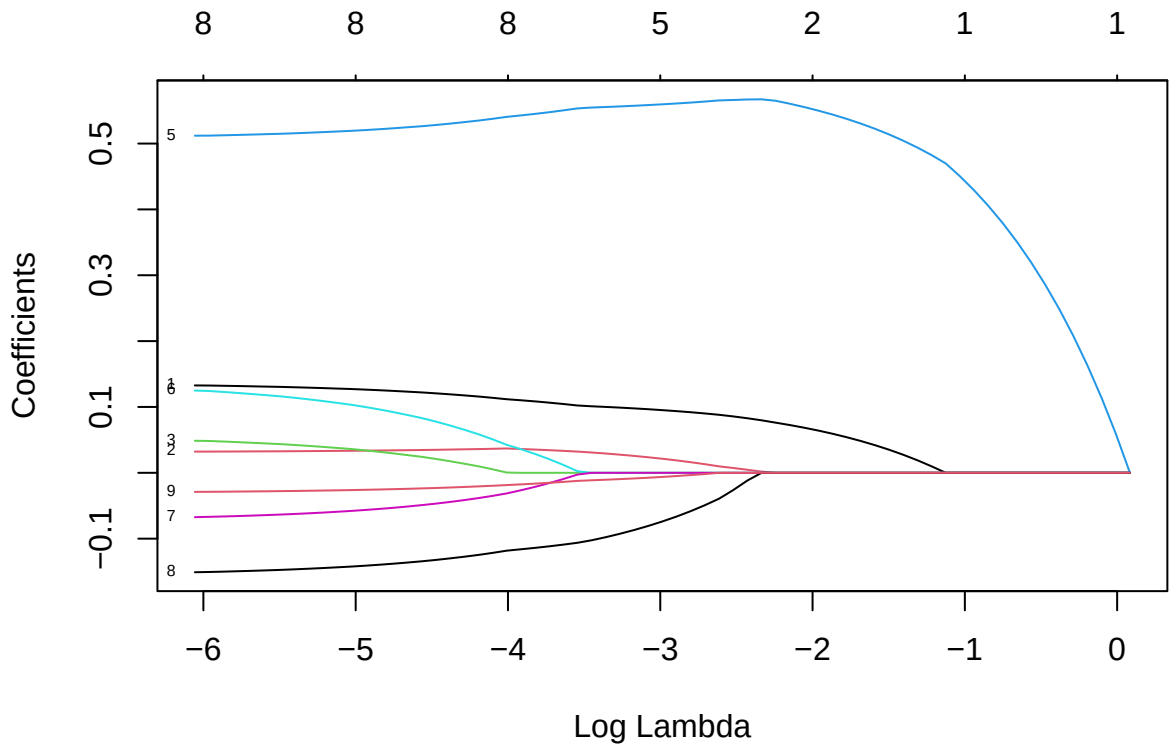
```
## [1] 0.007854436
```

Lambda value that minimizes 10-fold CV error = 0.007854436

(b) Plot the regression coefficient path.

```
#Regression Coefficient Plot
set.seed(100)

model.lasso = glmnet(as.matrix(trainData[,-10]), trainData[,10], family="gaussian", alpha=1, nfolds=10)
plot(model.lasso, xvar="lambda", label = T)
```



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

```
#Lasso Coefficients @ Lambda
set.seed(100)

coef(model.lasso, s = l.cv$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (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
```

8 variables were selected: nHM, piPC09, PCD, MLOGP, ON1V, N.072, B02.C.N., F04.C.O. X2Av was the single variable excluded.

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.

```
#Optimal Lambda - Elastic
set.seed(100)

e.cv = cv.glmnet(as.matrix(trainData[,-10]), trainData[,10], family="gaussian", alpha=0.5, nfolds=10)

e.cv$lambda.min
```

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

Lambda value that minimizes 10-fold CV error = 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?

```
#Elastic Coefficients @ Lambda
set.seed(100)

model.elastic = glmnet(as.matrix(trainData[,-10]), trainData[,10], family="gaussian", alpha=0.5, nfolds=

coef(model.elastic, s = e.cv$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (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
```

8 variables were selected. The 8 selected are the same 8 selected in the lasso regression.

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. Display the first few predictions for each model.

```
set.seed(100)
#Full Model Prediction
full = predict(model1, testData)

#Backwards Stepwise Model Prediction
backward = predict(model4, testData)

#Ridge Model Prediction
ridge = as.vector(predict(model.ridge, as.matrix(testData[,-10]), s =r.cv$lambda.min))

#Lasso Model Prediction
lasso = as.vector(predict(model.lasso, as.matrix(testData[,-10]), s = l.cv$lambda.min))

#Elastic Net Model Prediction
elastic = as.vector(predict(model.elastic, as.matrix(testData[,-10]), s = e.cv$lambda.min))

predictions = data.frame(logBCF = testData$logBCF, full, backward, ridge, lasso, elastic)

head(predictions, 5)
```

```
##     logBCF     full backward    ridge    lasso  elastic
## 714   2.64 2.446479 2.424916 2.454878 2.442895 2.441506
## 503   4.58 4.333759 4.353167 4.234425 4.313509 4.296451
## 358   3.44 3.266892 3.274192 3.223166 3.260617 3.252638
## 624   0.67 1.664770 1.297175 1.734094 1.610006 1.606774
## 718   2.61 1.955362 2.001399 1.993967 1.939946 1.939465
```

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

```
set.seed(100)
MSPE.full =mean((full-testData$logBCF)^2)
MSPE.backward = mean((backward-testData$logBCF)^2)
MSPE.ridge = mean((ridge-testData$logBCF)^2)
MSPE.lasso = mean((lasso-testData$logBCF)^2)
MSPE.elastic = mean((elastic - testData$logBCF)^2)

MSPE = data.frame(MSPE.full, MSPE.backward, MSPE.ridge, MSPE.lasso, MSPE.elastic)
MSPE
```

```
##   MSPE.full MSPE.backward MSPE.ridge MSPE.lasso MSPE.elastic
## 1 0.5839296     0.5742198  0.5877835  0.5790832     0.578275
```

The best model with respect to the mean squared prediction error is the backward stepwise regression model (model3). It has the lowest MSPE of any of the other models (MSPE=0.5742198), although by a relatively thin margin.

(c) Provide a table listing each method described in Question 7a and the variables selected by each method (see Lesson 5.8 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

Variables nHM, piPC09, MLOGP, F04.C.O. were consistently picked through all four methods.