Homework 5 Peer Assessment

Fall Semester 2020

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.

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
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, ]
```

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
```

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

significant at the 95% confidence level

```
coef<- which(summary(model1)$coefficients[,4]<0.05)
coef
```

```
##      nHM    MLOGP     ON1V B02.C.N. F04.C.O.
##        2        6        7        9       10
```

significant at the 99% confidence level

```
coef1<- which(summary(model1)$coefficients[,4]<0.01)
coef1
```

```
##      nHM    MLOGP F04.C.O.
##        2        6       10
```

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

```
set.seed(100)
n=nrow(trainData)
glmmodel1 <- glm(logBCF~.,data =trainData)
c(cv.glm(trainData, glmfit = glmmodel1, K=10)$delta[1], cv.glm(trainData, glmmodel1, K=n)$delta[
1])
```

```
## [1] 0.6512928 0.6529872
```

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

CP = 9.932584
AIC = 1497.476533
BIC = 1546.274187

```
set.seed(100)

c(Cp(model1,S2= 0.7957^2), AIC(model1, k=2),AIC(model1, k=log(n)))
```

```
## [1]    9.932584 1497.476533 1546.274187
```

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.

From the below ANOVA analysis we can see that p-value(0.00523) is small, hence we can reject the null hypothesis that the regression coefficients for piPC09,PCD,X2Av,ON1V,N.072 and B02.C.N. are not zero at alpha level or 0.05.
It is not a good practice to select variables based on statistical significance of individual coefficients because the significant depends on the other variables in the model. Once remove the insignificant variables from the model there are chances that other variables might become insignificant and vice-versa.

```
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
```

```
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
```

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.

There are 512 possible models. Below is the table which indicates the best model of each size.

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

```
##   1 2 3 4 5 6 7 8 9
## 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
```

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.

The best model is with 6 variables with Cp of 6.11

nHM
piPC09
MLOGP
ON1V
B02.C.N.
F04.C.O.

```
set.seed(100)
best.model = which(out$Cp==min(out$Cp))
cbind(as.matrix(out$which),out$Cp)[best.model,]
```

```
##        1        2        3        4        5        6        7        8
## 1.000000 1.000000 0.000000 0.000000 1.000000 1.000000 0.000000 1.000000
##        9
## 1.000000 6.116174
```

```
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*

```
set.seed(100)
full <- model1
n=nrow(trainData)
minimum <- lm(logBCF~1,data =trainData)
model4 <- step(full, scope=list(lower=minimum, upper=full), direction="backward", k = log(n))
```

```
## Start:  AIC=-231
## logBCF ~ nHM + piPC09 + PCD + X2Av + MLOGP + ON1V + N.072 + B02.C.N. +
##     F04.C.O.
##
##            Df Sum of Sq    RSS      AIC
## - X2Av      1     0.010 388.71 -237.417
## - PCD       1     0.481 389.18 -236.662
## - N.072     1     0.676 389.38 -236.350
## - piPC09    1     1.411 390.11 -235.173
## - B02.C.N.  1     2.468 391.17 -233.484
## - ON1V      1     2.804 391.51 -232.949
## <none>                  388.70 -230.997
## - F04.C.O.  1     6.410 395.11 -227.226
## - nHM       1    23.557 412.26 -200.718
## - MLOGP     1   138.539 527.24  -47.211
##
## Step:  AIC=-237.42
## logBCF ~ nHM + piPC09 + PCD + MLOGP + ON1V + N.072 + B02.C.N. +
##     F04.C.O.
##
##            Df Sum of Sq    RSS      AIC
## - PCD       1     0.517 389.23 -243.025
## - N.072     1     0.667 389.38 -242.783
## - piPC09    1     1.423 390.14 -241.574
## - B02.C.N.  1     2.510 391.22 -239.838
## - ON1V      1     2.915 391.63 -239.192
## <none>                  388.71 -237.417
## - F04.C.O.  1     6.491 395.21 -233.520
## - nHM       1    25.431 414.15 -204.309
## - MLOGP     1   146.081 534.80  -44.772
##
## Step:  AIC=-243.02
## logBCF ~ nHM + piPC09 + MLOGP + ON1V + N.072 + B02.C.N. + F04.C.O.
##
##            Df Sum of Sq    RSS      AIC
## - N.072     1     0.813 390.04 -248.159
## - B02.C.N.  1     2.099 391.33 -246.105
## - ON1V      1     2.412 391.64 -245.606
## <none>                  389.23 -243.025
## - F04.C.O.  1     6.088 395.32 -239.776
## - piPC09    1     6.203 395.43 -239.594
## - nHM       1    27.541 416.77 -206.800
## - MLOGP     1   181.833 571.06  -10.264
##
## Step:  AIC=-248.16
## logBCF ~ nHM + piPC09 + MLOGP + ON1V + B02.C.N. + F04.C.O.
##
##            Df Sum of Sq    RSS      AIC
## - ON1V      1     1.978 392.02 -251.438
## - B02.C.N.  1     3.026 393.07 -249.773
## <none>                  390.04 -248.159
## - piPC09    1     5.626 395.67 -245.659
```

```
## - F04.C.O.  1     5.851 395.89 -245.304
## - nHM       1    26.728 416.77 -213.236
## - MLOGP     1   203.819 593.86    7.728
##
## Step:  AIC=-251.44
## logBCF ~ nHM + piPC09 + MLOGP + B02.C.N. + F04.C.O.
##
##            Df Sum of Sq    RSS      AIC
## - B02.C.N.  1     2.693 394.72 -253.602
## - F04.C.O.  1     3.902 395.92 -251.695
## <none>                  392.02 -251.438
## - piPC09    1     7.252 399.27 -246.437
## - nHM       1    25.197 417.22 -219.003
## - MLOGP     1   247.006 639.03   47.031
##
## Step:  AIC=-253.6
## logBCF ~ nHM + piPC09 + MLOGP + F04.C.O.
##
##            Df Sum of Sq    RSS      AIC
## <none>                  394.72 -253.602
## - F04.C.O.  1     4.868 399.58 -252.390
## - piPC09    1     5.798 400.51 -250.939
## - nHM       1    26.847 421.56 -218.977
## - MLOGP     1   302.931 697.65   95.359
```

```
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?

There are 4 regression coefficients in model4.

```
coef4<- which(summary(model4)$coefficients[,4]<0.01)
coef4
```

```
##      nHM   piPC09    MLOGP F04.C.O.
##        2        3        4        5
```

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*?

Model 4 has only 4 variables while model5 has 6 variables.
“B02.C.N.” and “ON1V” are included in model5 and they are not significant at alpha level of 0.01.

```
set.seed(100)
model5 <- step(minimum, scope=list(lower=minimum, upper=full), direction="forward", k = 2)
```

```
## Start:  AIC=393.14
## logBCF ~ 1
##
##            Df Sum of Sq     RSS     AIC
## + MLOGP     1    738.32  429.60 -228.94
## + nHM       1    255.66  912.25  240.98
## + piPC09    1    220.90  947.02  264.31
## + PCD       1    150.75 1017.17  308.90
## + B02.C.N.  1    139.23 1028.68  315.93
## + N.072     1     43.55 1124.37  371.43
## + ON1V      1     27.76 1140.16  380.13
## + F04.C.O.  1     20.79 1147.13  383.93
## <none>                  1167.92  393.14
## + X2Av      1      2.45 1165.46  393.83
##
## Step:  AIC=-228.94
## logBCF ~ MLOGP
##
##            Df Sum of Sq    RSS     AIC
## + nHM       1   27.1327 402.47 -267.65
## + B02.C.N.  1    4.1778 425.42 -233.04
## + F04.C.O.  1    4.1526 425.45 -233.00
## + X2Av      1    3.2819 426.32 -231.72
## + ON1V      1    2.3664 427.23 -230.38
## <none>                  429.60 -228.94
## + piPC09    1    1.0443 428.55 -228.46
## + N.072     1    0.2481 429.35 -227.30
## + PCD       1    0.1198 429.48 -227.11
##
## Step:  AIC=-267.65
## logBCF ~ MLOGP + nHM
##
##            Df Sum of Sq    RSS     AIC
## + piPC09    1   2.88247 399.58 -270.13
## + F04.C.O.  1   1.95225 400.51 -268.68
## + B02.C.N.  1   1.93200 400.53 -268.65
## <none>                  402.47 -267.65
## + PCD       1   1.23679 401.23 -267.57
## + N.072     1   0.40989 402.06 -266.29
## + ON1V      1   0.33115 402.13 -266.16
## + X2Av      1   0.11836 402.35 -265.83
##
## Step:  AIC=-270.13
## logBCF ~ MLOGP + nHM + piPC09
##
##            Df Sum of Sq    RSS     AIC
## + F04.C.O.  1    4.8680 394.72 -275.78
## + B02.C.N.  1    3.6597 395.92 -273.88
## + N.072     1    1.4631 398.12 -270.42
## <none>                  399.58 -270.13
## + X2Av      1    0.5349 399.05 -268.97
## + ON1V      1    0.0065 399.58 -268.14
```

```
## + PCD       1    0.0001 399.58 -268.13
##
## Step:  AIC=-275.78
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O.
##
##            Df Sum of Sq    RSS     AIC
## + B02.C.N.  1   2.69326 392.02 -278.06
## + ON1V      1   1.64544 393.07 -276.39
## <none>                  394.72 -275.78
## + N.072     1   1.06163 393.65 -275.46
## + X2Av      1   0.51804 394.20 -274.60
## + PCD       1   0.07778 394.64 -273.91
##
## Step:  AIC=-278.06
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N.
##
##         Df Sum of Sq    RSS     AIC
## + ON1V   1   1.97807 390.04 -279.21
## <none>               392.02 -278.06
## + N.072  1   0.37905 391.64 -276.66
## + X2Av   1   0.12543 391.90 -276.25
## + PCD    1   0.00000 392.02 -276.06
##
## Step:  AIC=-279.21
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N. + ON1V
##
##         Df Sum of Sq    RSS     AIC
## <none>               390.04 -279.21
## + N.072  1   0.81306 389.23 -278.51
## + PCD    1   0.66238 389.38 -278.27
## + X2Av   1   0.02794 390.02 -277.26
```

```
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
```

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^2 As we can see in below table all the models have same "Adj. R-squared" but the number of predictors are different. Based on this I would recommend model4 as it has the less no of predictors among three models. Based on Cp, AIC and BIC I would choose model3 as it has lowest Cp, AIC and BIC value.

```
df <- data.frame("Adj R Squared" = c(round(summary(model1)$adj.r.squared,2),round(summary(model
3)$adj.r.squared,2),round(summary(model4)$adj.r.squared,2)), "No of Predictors" = c(length(coef
(model1))-1,length(coef(model3))-1,length(coef(model4))-1), "Model" = c("Model1", "Model3", "Mod
el4"))
rownames(df)<-df$Model
df$Model <- NULL
df
```

```
##        Adj.R.Squared No.of.Predictors
## Model1          0.66                9
## Model3          0.66                6
## Model4          0.66                4
```

**Cp, AIC and BIC,

```
c(Cp(model1,S2= 0.7957^2), AIC(model1, k=2),AIC(model1, k=log(n)))
```

```
## [1]    9.932584 1497.476533 1546.274187
```

```
c(Cp(model3,S2= 0.7957^2), AIC(model3, k=2),AIC(model3, k=log(n)))
```

```
## [1]    6.048526 1493.623474 1529.112677
```

```
c(Cp(model4,S2= 0.7957^2), AIC(model4, k=2),AIC(model1, k=log(n)))
```

```
## [1]    9.426582 1497.052364 1546.274187
```

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.

```
set.seed(100)
pred <- as.matrix(trainData[,-10])
ridge.cv<- cv.glmnet(pred, trainData$logBCF, alpha=0, nfolds=10)
ridge.cv
```

```
##
## Call:  cv.glmnet(x = pred, y = trainData$logBCF, nfolds = 10, alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure      SE Nonzero
## min 0.1088  0.6499 0.05413       9
## 1se 0.5805  0.6976 0.04453       9
```

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

```
set.seed(100)
ridge.model <- glmnet(pred, trainData$logBCF, alpha=0, nlambda = 100)
coef(ridge.model,ridge.cv$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.

The Ridge regression does not remove any variables hence there are all 9 variables in the output.

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)

lasso.cv<- cv.glmnet(pred, trainData$logBCF, alpha=1, nfolds=10)
lasso.cv
```

```
##
## Call:  cv.glmnet(x = pred, y = trainData$logBCF, nfolds = 10, alpha = 1)
##
## Measure: Mean-Squared Error
##
##      Lambda Measure      SE Nonzero
## min 0.00785  0.6482 0.05770       8
## 1se 0.18572  0.7029 0.05249       2
```

b. Plot the regression coefficient path.

```
set.seed(100)
lasso.model <- glmnet(pred, trainData$logBCF, alpha=1, nlambda = 100)

ls <- coef(lasso.model,lasso.cv$lambda.min)
ls
```

```
## 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
```

c. How many variables were selected? Which are they?

The Lasso regression selects all predictors except “X2AV”.
In total it selects 8 variables.

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)
elastic.cv<- cv.glmnet(pred, trainData$logBCF, alpha=0.5, nfolds=10)
elastic.cv
```

```
##
## Call:  cv.glmnet(x = pred, y = trainData$logBCF, nfolds = 10, alpha = 0.5)
##
## Measure: Mean-Squared Error
##
##      Lambda Measure      SE Nonzero
## min 0.02077  0.6478 0.05686       8
## 1se 0.25602  0.6968 0.05056       4
```

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 net also selects the 8 variables same as Lasso regression. They are very similar to variables selected in Lasso.

```
set.seed(100)
en.model <- glmnet(pred, trainData$logBCF, alpha=0.5, nlambda = 100)

en <- coef(en.model, elastic.cv$lambda.min)
en
```

```
## 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
```

```
ls
```

```
## 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
```

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.

Full Model, Backward Stepwiese and Ridge regression

```
set.seed(100)
x <- as.matrix(testData[,-10])
full.predict <- predict(model1, testData, interval = "prediction")
head(full.predict)
```

```
##          fit        lwr      upr
## 714 2.446479 0.87707975 4.015878
## 503 4.333759 2.76395272 5.903565
## 358 3.266892 1.69926856 4.834515
## 624 1.664770 0.06548615 3.264053
## 718 1.955362 0.38323014 3.527495
## 470 4.333278 2.76342774 5.903128
```

```
back.predict <- predict(model4, testData, interval = "prediction")
head(back.predict)
```

```
##          fit        lwr      upr
## 714 2.424916  0.8514428 3.998389
## 503 4.353167  2.7778585 5.928476
## 358 3.274192  1.7018148 4.846569
## 624 1.297175 -0.2817311 2.876082
## 718 2.001399  0.4265183 3.576279
## 470 4.355470  2.7801377 5.930802
```

```
ridge.predict <- predict(ridge.model, s= ridge.cv$lambda.min, newx = x)

head(ridge.predict)
```

```
##            1
## 714 2.454878
## 503 4.234425
## 358 3.223166
## 624 1.734094
## 718 1.993967
## 470 4.233222
```

LAssso regression

```
lasso.pred <- as.matrix(x[,-4])
newlasso <- glmnet(lasso.pred, testData$logBCF, alpha=1, nlambda = 100)
lasso.predict <- predict(newlasso,s= lasso.cv$lambda.min, newx = lasso.pred )
head(lasso.predict)
```

```
##            1
## 714 2.476987
## 503 4.361715
## 358 3.311588
## 624 1.202483
## 718 2.014811
## 470 4.365470
```

elastic net prediction

```
newen <- glmnet(lasso.pred, testData$logBCF, alpha=0.5, nlambda = 100)
en.predict <- predict(newen,s= lasso.cv$lambda.min, newx = lasso.pred )
head(en.predict)
```

```
##           1
## 714 2.485301
## 503 4.371947
## 358 3.320561
## 624 1.189570
## 718 2.024125
## 470 4.375843
```

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

Based on MSPE we can say that the Elastic net model outperforms all other models.

```
set.seed(100)
cat("MSPE of fullmodel:", mean((full.predict[,1]-testData$logBCF)^2), end="\n")
```

```
## MSPE of fullmodel: 0.5839296
```

```
cat("MSPE of backward:", mean((back.predict[,1]-testData$logBCF)^2), end="\n")
```

```
## MSPE of backward: 0.5742198
```

```
cat("MSPE of Ridge:", mean((ridge.predict-testData$logBCF)^2), end="\n")
```

```
## MSPE of Ridge: 0.5877835
```

```
cat("MSPE of Lasso:", mean((lasso.predict-testData$logBCF)^2), end="\n")
```

```
## MSPE of Lasso: 0.55096
```

```
cat("MSPE of Elasticnet:", mean((en.predict-testData$logBCF)^2), end="\n")
```

```
## MSPE of Elasticnet: 0.5507579
```

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?

From the below table we can say that Backward regression has the lowest number of variables and those variables are selected constantly by all other models as well.

	Backward Stepwise	**Ridge**	**Lasso**
nHM	yes	yes	yes
piPC09	yes	yes	yes
PCD	-	yes	yes

	Backward Stepwise	Ridge	Lasso
X2AV	-	yes	-
MLOGP	yes	yes	yes
ON1V	-	yes	yes
N.072	-	yes	yes
B02.C.N.	-	yes	yes
F04.C.O.	yes	yes	yes