Homework 4 Peer Assessment

Summer 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.

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)
```

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

```r
attach(trainData)
```

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.

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

```
##
## Call:
## glm(formula = logBCF ~ ., data = trainData)
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 0.633069)
##
##     Null deviance: 1167.9  on 623  degrees of freedom
## Residual deviance:  388.7  on 614  degrees of freedom
## AIC: 1497.5
##
## Number of Fisher Scoring iterations: 2
```

```r
n=nrow(trainData)
```

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

```r
# regression coefficients are significant at the 95% confidence level
which(summary(model1)$coeff[,4]<=0.05)
```

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

```r
# regression coefficients are significant at the 99% confidence level
which(summary(model1)$coeff[,4]<=0.01)
```

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

Answer :

1. *95% - nHM, MLOGP, ON1V, B02.C.N. and F04.C.O.*
2. *99% - nHM, MLOGP and F04.C.O.*

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

```r
set.seed(100)
#10-Fold
m1.10.fold = cv.glm(trainData, model1, K=10)
(m1.10.fold)$delta
```

```
## [1] 0.6512928 0.6497704
```

```r
#leave one out cross-validation
#m1.loocv = cv.glm(trainData, model1, K=n)
m1.loocv = cv.glm(trainData, model1, K=nrow(trainData))
(m1.loocv)$delta
```

```
## [1] 0.6529872 0.6529625
```

```r
#c(cv.glm(trainData, model1, K=10)$delta, cv.glm(trainData, model1, K=n)$delta)
#cv.glm(trainData, model1, K=10)
```

Answer :

- 10-Fold - 0.6512928 0.6497704
- leave one out cross-validation - 0.6529872 0.6529625

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

```r
set.seed(100)
cat("Mallow's Cp - ", Cp(model1, S2=sigma(model1)^2), "\n")
```

```
## Mallow's Cp -  10
```

```r
cat("AIC         - ", AIC(model1, k=2), "\n")
```

```
## AIC         -  1497.477
```

```r
cat("BIC         - ", AIC(model1, k=log(n)), "\n")
```

```
## BIC         -  1546.274
```

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

Answer :

- Mallow's Cp - 10.000
- AIC - 1497.477
- BIC - 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.

```r
set.seed(100)
model2 = glm(logBCF~ nHM+MLOGP+F04.C.O., data = trainData)
summary(model2)
```

```
##
## Call:
## glm(formula = logBCF ~ nHM + MLOGP + F04.C.O., data = trainData)
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 0.6459897)
##
##     Null deviance: 1167.92  on 623  degrees of freedom
## Residual deviance:  400.51  on 620  degrees of freedom
## AIC: 1504.2
##
## Number of Fisher Scoring iterations: 2
```

```r
a1 = anova(model2, model1, test = "F")
a1
```

```
## Analysis of Deviance 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.
##   Resid. Df Resid. Dev Df Deviance     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
```

```r
1-pchisq( abs(a1$Deviance[2]), abs(a1$Df[2]))
```

```
## [1] 0.06636137
```

It is not a good idea to select variables based on statistical significance.. in the full model F04.C.O was significant, but in the reduced model it is no longer significant. I would prefer the Full model as the AIC value is lower and the model fits better. Alo the p-Value is 0.005 … so atleast one of the variables in model 1 that is not in model 2 is statistically significant

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.

Answer : There are total of $2^9 = 512$ possible models.

Hint: You can use nbest parameter.

```
set.seed(100)
out = leaps(trainData[,-c(10)], logBCF, method = "Cp", nbest = 1, names = colnames(trainData[,-c(10)]))
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
```

(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.

```
set.seed(100)

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
```

```
model3 = glm(logBCF~ nHM+piPC09+MLOGP+ON1V+B02.C.N.+F04.C.O., data = trainData)
summary(model3)
```

```
##
## Call:
## glm(formula = logBCF ~ nHM + piPC09 + MLOGP + ON1V + B02.C.N. +
##     F04.C.O., data = trainData)
##
## Deviance 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
##
```

```
## (Dispersion parameter for gaussian family taken to be 0.6321621)
##
##     Null deviance: 1167.92  on 623  degrees of freedom
## Residual deviance:  390.04  on 617  degrees of freedom
## AIC: 1493.6
##
## Number of Fisher Scoring iterations: 2
```

Answer : There are 6 variables in the model with lowest Mallow's Cp value. The columns for the best model are - nHM, piPC09, MLOGP, ON1V, B02.C.N. & 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)
bstep = step(model1, direction="backward", k = log(n))
```

```
## Start:  AIC=1541.84
## logBCF ~ nHM + piPC09 + PCD + X2Av + MLOGP + ON1V + N.072 + B02.C.N. +
##     F04.C.O.
##
##            Df Deviance    AIC
## - X2Av      1   388.71 1535.4
## - PCD       1   389.18 1536.2
## - N.072     1   389.38 1536.5
## - piPC09    1   390.11 1537.7
## - B02.C.N.  1   391.17 1539.3
## - ON1V      1   391.51 1539.9
## <none>          388.70 1541.8
## - F04.C.O.  1   395.11 1545.6
## - nHM       1   412.26 1572.1
## - MLOGP     1   527.24 1725.6
##
## Step:  AIC=1535.42
## logBCF ~ nHM + piPC09 + PCD + MLOGP + ON1V + N.072 + B02.C.N. +
##     F04.C.O.
##
##            Df Deviance    AIC
## - PCD       1   389.23 1529.8
## - N.072     1   389.38 1530.0
## - piPC09    1   390.14 1531.3
## - B02.C.N.  1   391.22 1533.0
## - ON1V      1   391.63 1533.6
## <none>          388.71 1535.4
## - F04.C.O.  1   395.21 1539.3
## - nHM       1   414.15 1568.5
## - MLOGP     1   534.80 1728.1
##
## Step:  AIC=1529.81
## logBCF ~ nHM + piPC09 + MLOGP + ON1V + N.072 + B02.C.N. + F04.C.O.
##
##            Df Deviance    AIC
```

```
## - N.072     1   390.04 1524.7
## - B02.C.N.  1   391.33 1526.7
## - ON1V      1   391.64 1527.2
## <none>          389.23 1529.8
## - F04.C.O.  1   395.32 1533.1
## - piPC09    1   395.43 1533.2
## - nHM       1   416.77 1566.0
## - MLOGP     1   571.06 1762.6
##
## Step:  AIC=1524.68
## logBCF ~ nHM + piPC09 + MLOGP + ON1V + B02.C.N. + F04.C.O.
##
##            Df Deviance    AIC
## - ON1V      1   392.02 1521.4
## - B02.C.N.  1   393.07 1523.1
## <none>          390.04 1524.7
## - piPC09    1   395.67 1527.2
## - F04.C.O.  1   395.89 1527.5
## - nHM       1   416.77 1559.6
## - MLOGP     1   593.86 1780.6
##
## Step:  AIC=1521.4
## logBCF ~ nHM + piPC09 + MLOGP + B02.C.N. + F04.C.O.
##
##            Df Deviance    AIC
## - B02.C.N.  1   394.72 1519.2
## - F04.C.O.  1   395.92 1521.1
## <none>          392.02 1521.4
## - piPC09    1   399.27 1526.4
## - nHM       1   417.22 1553.8
## - MLOGP     1   639.03 1819.9
##
## Step:  AIC=1519.23
## logBCF ~ nHM + piPC09 + MLOGP + F04.C.O.
##
##            Df Deviance    AIC
## <none>          394.72 1519.2
## - F04.C.O.  1   399.58 1520.5
## - piPC09    1   400.51 1521.9
## - nHM       1   421.56 1553.9
## - MLOGP     1   697.65 1868.2
```

```
summary(bstep)
```

```
##
## Call:
## glm(formula = logBCF ~ nHM + piPC09 + MLOGP + F04.C.O., data = trainData)
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 0.6376662)
##
##     Null deviance: 1167.92  on 623  degrees of freedom
## Residual deviance:  394.72  on 619  degrees of freedom
## AIC: 1497.1
##
## Number of Fisher Scoring iterations: 2
```

```
bstep$anova
```

```
##         Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1            NA         NA       614   388.7043 1541.838
## 2     - X2Av  1 0.01001271       615   388.7144 1535.418
## 3      - PCD  1 0.51660710       616   389.2310 1529.811
## 4    - N.072  1 0.81306408       617   390.0440 1524.677
## 5     - ON1V  1 1.97807463       618   392.0221 1521.397
## 6 - B02.C.N.  1 2.69325705       619   394.7154 1519.233
```

```
bstep$coefficients
```

```
##  (Intercept)          nHM       piPC09        MLOGP     F04.C.O.
## -0.008695283  0.114029425  0.041119211  0.566473195 -0.022103780
```

```
bstep$formula
```

```
## logBCF ~ nHM + piPC09 + MLOGP + F04.C.O.
```

```
model4 = glm(logBCF ~ nHM + piPC09 + MLOGP + F04.C.O., data = trainData)
summary(model4)
```

```
##
## Call:
## glm(formula = logBCF ~ nHM + piPC09 + MLOGP + F04.C.O., data = trainData)
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 0.6376662)
##
```

```
##     Null deviance: 1167.92  on 623  degrees of freedom
## Residual deviance:  394.72  on 619  degrees of freedom
## AIC: 1497.1
##
## Number of Fisher Scoring iterations: 2
```

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

```
which(summary(model4)$coeff[,4]<=0.01)
```

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

Answer : There are 4 variables in *model4*. All the regression coefficients (nHM, piPC09, MLOGP & F04.C.O.) are significant at 99% CI.

(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*?

```
set.seed(100)

fstep = step(glm(logBCF ~ 1), scope=list(upper=model1), direction="forward")
```

```
## Start:  AIC=2165.97
## logBCF ~ 1
##
##            Df Deviance    AIC
## + MLOGP     1   429.60 1543.9
## + nHM       1   912.25 2013.8
## + piPC09    1   947.02 2037.2
## + PCD       1  1017.17 2081.7
## + B02.C.N.  1  1028.68 2088.8
## + N.072     1  1124.37 2144.3
## + ON1V      1  1140.16 2153.0
## + F04.C.O.  1  1147.13 2156.8
## <none>         1167.92 2166.0
## + X2Av      1  1165.46 2166.7
##
## Step:  AIC=1543.9
## logBCF ~ MLOGP
##
##            Df Deviance    AIC
## + nHM       1   402.47 1505.2
## + B02.C.N.  1   425.42 1539.8
## + F04.C.O.  1   425.45 1539.8
## + X2Av      1   426.32 1541.1
## + ON1V      1   427.23 1542.5
## <none>          429.60 1543.9
## + piPC09    1   428.55 1544.4
## + N.072     1   429.35 1545.5
## + PCD       1   429.48 1545.7
##
## Step:  AIC=1505.19
## logBCF ~ MLOGP + nHM
##
```

```
##             Df Deviance    AIC
## + piPC09    1   399.58 1502.7
## + F04.C.O.  1   400.51 1504.2
## + B02.C.N.  1   400.53 1504.2
## <none>          402.47 1505.2
## + PCD       1   401.23 1505.3
## + N.072     1   402.06 1506.5
## + ON1V      1   402.13 1506.7
## + X2Av      1   402.35 1507.0
##
## Step:  AIC=1502.7
## logBCF ~ MLOGP + nHM + piPC09
##
##             Df Deviance    AIC
## + F04.C.O.  1   394.72 1497.0
## + B02.C.N.  1   395.92 1499.0
## + N.072     1   398.12 1502.4
## <none>          399.58 1502.7
## + X2Av      1   399.05 1503.9
## + ON1V      1   399.58 1504.7
## + PCD       1   399.58 1504.7
##
## Step:  AIC=1497.05
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O.
##
##             Df Deviance    AIC
## + B02.C.N.  1   392.02 1494.8
## + ON1V      1   393.07 1496.5
## <none>          394.72 1497.0
## + N.072     1   393.65 1497.4
## + X2Av      1   394.20 1498.2
## + PCD       1   394.64 1498.9
##
## Step:  AIC=1494.78
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N.
##
##          Df Deviance    AIC
## + ON1V   1   390.04 1493.6
## <none>       392.02 1494.8
## + N.072  1   391.64 1496.2
## + X2Av   1   391.90 1496.6
## + PCD    1   392.02 1496.8
##
## Step:  AIC=1493.62
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N. + ON1V
##
##          Df Deviance    AIC
## <none>       390.04 1493.6
## + N.072  1   389.23 1494.3
## + PCD    1   389.38 1494.6
## + X2Av   1   390.02 1495.6
```

```
summary(fstep)
```

```
##
```

```
## Call:
## glm(formula = logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N. +
##     ON1V)
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 0.6321621)
##
##     Null deviance: 1167.92  on 623  degrees of freedom
## Residual deviance:  390.04  on 617  degrees of freedom
## AIC: 1493.6
##
## Number of Fisher Scoring iterations: 2
```

```
fstep$anova
```

```
##         Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1            NA         NA       623  1167.9156 2165.974
## 2    + MLOGP -1 738.316995       622   429.5986 1543.897
## 3      + nHM -1  27.132735       621   402.4659 1505.186
## 4   + piPC09 -1   2.882474       620   399.5834 1502.701
## 5 + F04.C.O. -1   4.868038       619   394.7154 1497.052
## 6 + B02.C.N. -1   2.693257       618   392.0221 1494.780
## 7     + ON1V -1   1.978075       617   390.0440 1493.623
```

```
fstep$coefficients
```

```
## (Intercept)       MLOGP         nHM      piPC09    F04.C.O.    B02.C.N.
##  0.03578462  0.52852233  0.12408604  0.04216683 -0.02864445 -0.16020422
##        ON1V
##  0.09809870
```

```
fstep$formula
```

```
## logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N. + ON1V
```

```
model5=glm(formula = logBCF ~ MLOGP + nHM + piPC09 + F04.C.O. + B02.C.N. + ON1V)
summary(model5)
```

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

```
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 0.6321621)
##
##     Null deviance: 1167.92  on 623  degrees of freedom
## Residual deviance:  390.04  on 617  degrees of freedom
## AIC: 1493.6
##
## Number of Fisher Scoring iterations: 2
```

```
coef(model4)
```

```
##  (Intercept)          nHM       piPC09        MLOGP     F04.C.O.
## -0.008695283  0.114029425  0.041119211  0.566473195 -0.022103780
```

```
coef(model5)
```

```
## (Intercept)       MLOGP         nHM      piPC09    F04.C.O.    B02.C.N.
##  0.03578462  0.52852233  0.12408604  0.04216683 -0.02864445 -0.16020422
##        ON1V
##  0.09809870
```

Answer : Variables included in model 5(*MLOGP, nHM, piPC09, F04.C.O., B02.C.N., ON1V*) are different from that in model 4 (*nHM, piPC09, MLOGP, F04.C.O.*)

(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?

```
set.seed(100)
#R-Squared
library(rsq)
rsq(model1,adj=TRUE,type="sse")
```

```
## [1] 0.6623027
```

```
rsq(model3,adj=TRUE,type="sse")
```

```
## [1] 0.6627864
```

```
rsq(model4,adj=TRUE,type="sse")
```

```
## [1] 0.6598504
```

```r
# with(summary(model1), 1 - deviance/null.deviance)
# with(summary(model3), 1 - deviance/null.deviance)
# with(summary(model4), 1 - deviance/null.deviance)
```

```r
#AIC
summary(model1)$aic
```

```
## [1] 1497.477
```

```r
summary(model3)$aic
```

```
## [1] 1493.623
```

```r
summary(model4)$aic
```

```
## [1] 1497.052
```

```r
#BIC
AIC(model1, k=log(n))
```

```
## [1] 1546.274
```

```r
AIC(model3, k=log(n))
```

```
## [1] 1529.113
```

```r
AIC(model4, k=log(n))
```

```
## [1] 1523.669
```

```r
# (log(n)/2) * summary(model1)$aic
# (log(n)/2) * summary(model3)$aic
# (log(n)/2) * summary(model4)$aic
```

```r
c(Cp(model1, S2=sigma(model1)^2), length(model1$coefficients) - 1)
```

```
## [1] 10  9
```

```r
c(Cp(model3, S2=sigma(model1)^2), length(model3$coefficients) - 1)
```

```
## [1] 6.116174 6.000000
```

```r
c(Cp(model4, S2=sigma(model1)^2), length(model4$coefficients) - 1)
```

```
## [1] 9.495041 4.000000
```

Answer : Based on

- R^2 - Model 3
- AIC - Model 3
- BIC - Model 4
- Mallow Cp - Model 3

Based on these model 3 is the best across all the 4 metrics

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.

```r
set.seed(100)
lambda = seq(0, 10, by=1)
```

```r
x.train <- model.matrix(logBCF ~ ., trainData)[,-1]
y.train <- logBCF

ridge.cv = cv.glmnet(x.train, y.train, alpha=0, nfolds = 10)
ridge = glmnet(x.train, y.train, alpha=0, nlambda=100)

ridge.cv$lambda.min
```

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

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

```r
set.seed(100)
# which(out$GCV == min(out$GCV))
# round(out$coef[,which(out$GCV == min(out$GCV))], 4)
#
# length(out$coef[,10])

coef(ridge, s=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.

Answer : All variables selected. But Ridge regression does not help in Variable selection.. it only shrinks the coefficients of correlated variables to 0

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)
# Xpred = cbind(nHM,piPC09,PCD,X2Av,MLOGP,ON1V,N.072,B02.C.N.,F04.C.O.)
x.train <- model.matrix(logBCF ~ ., trainData)[,-1]
y.train <- logBCF

lasso.cv = cv.glmnet(x.train, y.train, alpha=1, nfolds=10)
lasso = glmnet(x.train, y.train, alpha=1, nlambda=100)
cat('Min Lambda : ', lasso.cv$lambda.min, '\n')
```

```
## Min Lambda :  0.007854436
```

```r
ccc = as.matrix(coef(lasso, s=lasso.cv$lambda.min))
ccc[ccc!=0,]
```

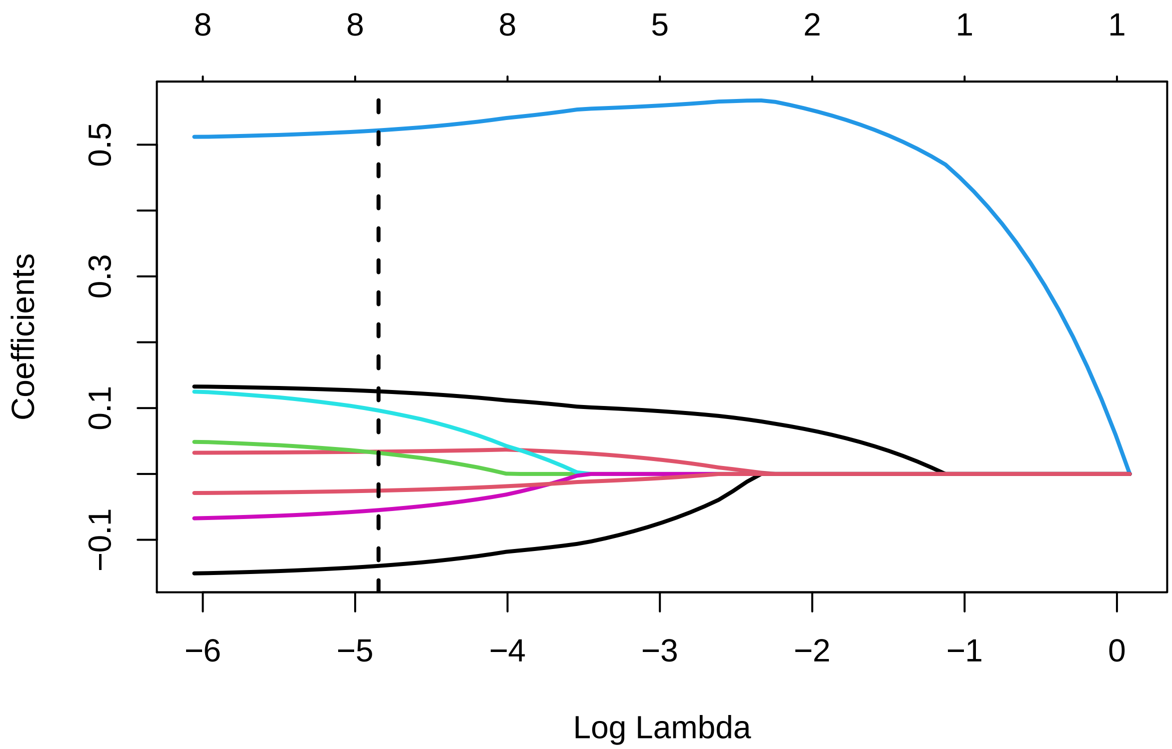
```
## (Intercept)          nHM       piPC09          PCD        MLOGP         ON1V
```

```
## 0.02722838 0.12543866 0.03387665 0.03194878 0.52174346 0.09633951
##      N.072    B02.C.N.    F04.C.O.
## -0.05487196 -0.13961811 -0.02535576
```

(b) Plot the regression coefficient path.

```r
set.seed(100)

plot(lasso, xvar="lambda", lwd=2)
abline(v=log(lasso.cv$lambda.min), col='black', lty=2, lwd=2)
```



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

8 Variables Selected - nHM,piPC09,PCD,MLOGP,ON1V,N.072,B02.C.N.,F04.C.O.

```r
set.seed(100)
index.lasso <- which(coef(lasso, lasso.cv$lambda.min) != 0)
cat("\nVariables selected by lasso regression: ",
    names(coef(model1)[index.lasso])[-c(1)], "\n")
```

```
##
## Variables selected by lasso regression:  nHM piPC09 PCD MLOGP ON1V N.072 B02.C.N. F04.C.O.
```

```r
cat("Numbers Variables selected", length(names(coef(model1)[index.lasso])) - 1, "\n")
```

```
## Numbers Variables selected 8
```

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.

```r
set.seed(100)
elanet.cv = cv.glmnet(x.train, y.train, alpha=0.5, nfolds=10)
```

```
elanet = glmnet(x.train, y.train, alpha=0.5, nlambda = 100)
```

(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)
elanet.cv$lambda.min
```

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

```
# plot(satmodel, xvar="lambda", lwd=2)
# abline(v=log(satmodel.cv$lambda.min), col='black', lty=2, lwd=2)
index.elanet <- which(coef(elanet, elanet.cv$lambda.min) != 0)
cat("\nVariables selected by Elastic Net regression : ",
    names(coef(model1)[index.elanet])[-c(1)], "\n")
```

```
##
## Variables selected by Elastic Net regression :  nHM piPC09 PCD MLOGP ON1V N.072 B02.C.N. F04.C.O.
```

```
cat("Numbers Variables selected : ", length(names(coef(model1)[index.elanet])) - 1, "\n")
```

```
## Numbers Variables selected :  8
```

Answer : There are 8 variabes selected. Both selected 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.

```
set.seed(100)
# Full Model
fullmodel.predict = predict(model1, newdata = testData)
head(fullmodel.predict, 3)
```

```
##      714      503      358
## 2.446479 4.333759 3.266892
```

```
# backward stepwise regression with BIC
bstep.bic.predict = predict(model4, newdata = testData)
head(bstep.bic.predict, 3)
```

```
##      714      503      358
## 2.424916 4.353167 3.274192
```

```
# # ridge regression
# as.matrix(cbind(const=1,testData)) %*% coef(out)
new_test <- model.matrix(logBCF ~ ., testData)[,-1]
# Obtain predicted probabilities for the test set
pred.ridge = predict(ridge, newx = new_test, s=ridge.cv$lambda.min)
head(pred.ridge, 3)
```

```
##            1
## 714 2.454878
## 503 4.234425
## 358 3.223166
```

```
# lasso regression
pred.lasso = predict(lasso, newx = new_test, s=lasso.cv$lambda.min)
head(pred.lasso, 3)
```

```
##             1
## 714 2.442895
## 503 4.313509
## 358 3.260617
```

```
# elastic net
pred.elnet = as.vector(predict(elanet, newx = new_test,s = elanet.cv$lambda.min))
head(pred.elnet, 3)
```

```
## [1] 2.441506 4.296451 3.252638
```

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

```
set.seed(100)
MSE_full <- mean((fullmodel.predict - testData$logBCF)^2)
MSE_full
```

```
## [1] 0.5839296
```

```
MSE_bstep <- mean((bstep.bic.predict - testData$logBCF)^2)
MSE_bstep
```

```
## [1] 0.5742198
```

```
MSE_ridge <- mean((pred.ridge - testData$logBCF)^2)
MSE_ridge
```

```
## [1] 0.5877835
```

```
MSE_lasso <- mean((pred.lasso - testData$logBCF)^2)
MSE_lasso
```

```
## [1] 0.5790832
```

```
MSE_elnem <- mean((pred.elnet - testData$logBCF)^2)
MSE_elnem
```

```
## [1] 0.578275
```

Answer : Best Model - Backward Stepwise with *BIC*.. model4

(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?

```
coef(bstep)
```

```
##  (Intercept)          nHM       piPC09        MLOGP     F04.C.O.
## -0.008695283  0.114029425  0.041119211  0.566473195 -0.022103780
```

```
out$coef[,which(out$GCV == min(out$GCV))]
```

```
## Warning in min(out$GCV): no non-missing arguments to min; returning Inf
```

```
## NULL
```

```
names(coef(model1)[index.lasso])[-c(1)]
```

```
## [1] "nHM"      "piPC09"   "PCD"      "MLOGP"    "ON1V"     "N.072"    "B02.C.N."
## [8] "F04.C.O."
```

```
names(coef(model1)[index.elanet])[-c(1)]
```

```
## [1] "nHM"      "piPC09"   "PCD"      "MLOGP"    "ON1V"     "N.072"    "B02.C.N."
## [8] "F04.C.O."
```

	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