STAT 331 FINAL PROJECT APPENDIX

library("faraway")  
protein\_train <- read.csv('protein-train.csv')  
  
##EDA  
  
# check for NAs  
na.list = sapply(protein\_train, function(x){sum(is.na(x))})  
na.list[na.list>0]

## named integer(0)

#summary of original dataset  
  
  
#summary(protein\_train)  
  
# Scatter plots  
par ( mfrow = c(3 ,3))  
plot ( protein\_train$angles , protein\_train$accuracy , ylab ="Accuracy", xlab ="Angles")  
plot ( protein\_train$carbonylC\_aliph2HC\_short , protein\_train$accuracy , ylab ="Accuracy", xlab =" carbonylC\_aliph2HC\_short")  
plot ( protein\_train$carbonylC\_bbCA\_medlong , protein\_train$accuracy , ylab ="Accuracy", xlab =" carbonylC\_bbCA\_medlong")  
plot ( protein\_train$scAGN\_bbC\_medshort , protein\_train$accuracy , ylab ="Accuracy", xlab =" scAGN\_bbC\_medshort")  
plot ( protein\_train$scArgN\_carboxylO\_long , protein\_train$accuracy , ylab ="Accuracy", xlab =" scArgN\_carboxylO\_long")  
plot ( protein\_train$bbC\_bbO\_vlong , protein\_train$accuracy , ylab ="Accuracy", xlab =" bbC\_bbO\_vlong")  
plot ( protein\_train$aliph2HC\_aliph3HC\_vshort , protein\_train$accuracy , ylab ="Accuracy", xlab =" aliph2HC\_aliph3HC\_vshort")  
  
mfull <- lm(accuracy~., data = protein\_train)  
  
  
summary(mfull)

# remove the 2 variables which were perfectly correlated together and create a new dataframe  
protein = subset(protein\_train, select = -c(scArgN\_bbC\_medshort,scArgN\_bbO\_short) )  
  
# correlation with accuracy  
  
cor.list = sapply(protein[-1], function(x){cor(protein$accuracy,x)})  
  
cor.tab = data.frame(var = names(cor.list), r=cor.list,row.names = NULL)  
  
cor.tab2 = subset(cor.tab, abs(r)>0.05)  
  
vars = cor.tab2$var  
  
vars = as.character(vars)  
protein1 = protein[ c("accuracy",as.character(vars))]  
  
# new linear model based on the new dataframe  
mshort<- lm(accuracy ~., data=protein1)  
summary(mshort)

max(vif(mshort))

## [1] 662.2596

# while loop to go through all the vifs, and only keep those variables with a vif < 10  
cutoff <- 10  
flag <- TRUE  
while(flag){  
 fit <- lm(accuracy ~ ., data=protein1)  
 vfit <- vif(fit)  
 if(max(vfit) > cutoff){  
 protein1 <- subset(protein1, select = -get(names(vfit)[which.max(vfit)]))  
 } else {  
 flag <- FALSE  
 }  
}  
  
#print(fit)  
#print(vfit)  
  
  
##MODEL SELECTION   
set.seed(20680907)  
library(MASS)  
  
#start with one train/validation split  
N <- nrow(protein1)  
trainInd <- sample(1:N, round(N\*0.8), replace=F)  
trainSet <- protein1[trainInd,]  
validSet <- protein1[-trainInd,]  
  
# Full model and empty model with just intercept  
full <- lm(accuracy ~ ., data = trainSet)  
empty <- lm(accuracy ~ 1, data =trainSet)  
  
# Stepwise forward with BIC  
#m1<-stepAIC(object = empty, scope = list(upper = full, lower = empty), direction = "forward", k = log(nrow(trainSet)))  
  
#save(m1, file= "m1.rda")  
load("m1.rda")  
summary(m1)

BIC(m1)

## [1] 3024.507

pred1 <- predict(m1, newdata = validSet)  
sqrt(mean((validSet$accuracy - pred1)^2)) # RMSE on validation

## [1] 0.6188801

sqrt(mean(m1$residuals^2)) # RMSE on train

## [1] 0.526641

# forward stepwise again, with a larger L0 penalty (e.g., twice the usual BIC penalty)  
#m2 <- stepAIC(object = empty, scope = list(upper = full, lower = empty), direction = "forward", k = 2\*log(nrow(trainSet)))  
  
#save(m2, file= "m2.rda")  
load("m2.rda")  
summary(m2)

BIC(m2)

## [1] 3230.409

pred2 <- predict(m2, newdata = validSet)  
sqrt(mean((validSet$accuracy - pred2)^2)) # RMSE on validation

## [1] 0.6570954

sqrt(mean(m2$residuals^2)) # RMSE on train

## [1] 0.6212791

#Forward-Backward  
#m3 <- stepAIC(object = empty, scope = list(upper = full, lower = empty), direction = "both", k = log(nrow(trainSet)))  
  
#save(m3, file= "m3.rda")  
load("m3.rda")  
summary(m3)

BIC(m3)

## [1] 3011.712

pred3 <- predict(m3, newdata = validSet)  
sqrt(mean((validSet$accuracy - pred3)^2)) # RMSE on validation

## [1] 0.6229044

sqrt(mean(m3$residuals^2)) # RMSE on train

## [1] 0.5332196

#Forward-Backward with larger penalty   
#m4 <- stepAIC(object = empty, scope = list(upper = full, lower = empty), direction = "both", k = 2\*log(nrow(trainSet)))  
  
#save(m4, file= "m4.rda")  
load("m4.rda")  
summary(m4)

BIC(m4)

## [1] 3220.17

pred4 <- predict(m4, newdata = validSet)  
sqrt(mean((validSet$accuracy - pred4)^2)) # RMSE on validation

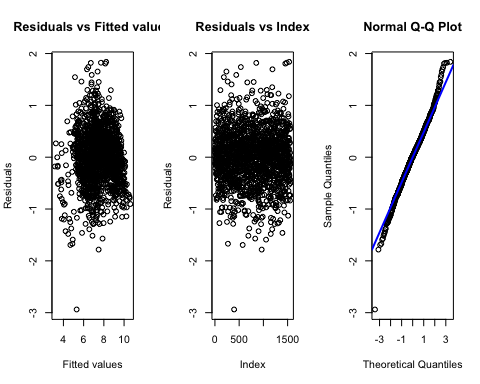
## [1] 0.682042

sqrt(mean(m4$residuals^2)) # RMSE on train

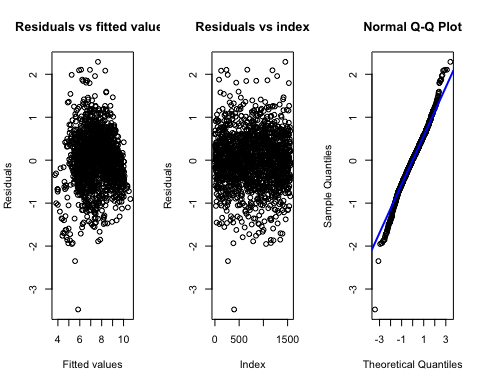
## [1] 0.6207029

#Plots of Models to Check Model Assumptions   
  
#m1  
par(mfrow=c(1,3))

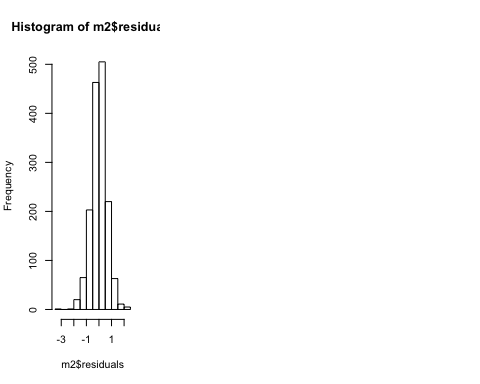
plot(m1$fitted.values,m1$residuals, xlab="Fitted values",  
ylab="Residuals", main="Residuals vs Fitted values")  
plot(1:nrow(trainSet),m1$residuals, xlab="Index",  
ylab="Residuals", main="Residuals vs Index")  
qqnorm(m1$residuals)  
qqline(m1$residuals, col="blue", lwd=2)



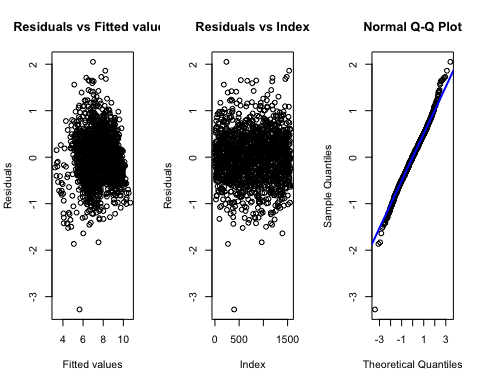
#m2  
plot(m2$fitted.values,m2$residuals, xlab="Fitted values",  
ylab="Residuals", main="Residuals vs fitted values")  
plot(1:nrow(trainSet),m2$residuals, xlab="Index",  
ylab="Residuals", main="Residuals vs index")  
qqnorm(m2$residuals)  
qqline(m2$residuals, col="blue", lwd=2)



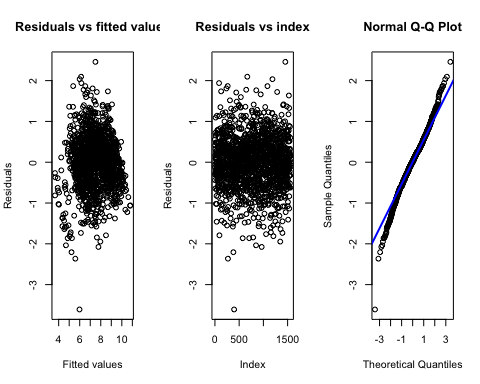
hist ( m2$residuals )  
  
#m3  
par(mfrow=c(1,3))



plot(m3$fitted.values,m3$residuals, xlab="Fitted values",  
ylab="Residuals", main="Residuals vs Fitted values")  
plot(1:nrow(trainSet),m3$residuals, xlab="Index",  
ylab="Residuals", main="Residuals vs Index")  
qqnorm(m3$residuals)  
qqline(m3$residuals, col="blue", lwd=2)



#m4  
plot(m4$fitted.values,m4$residuals, xlab="Fitted values",  
ylab="Residuals", main="Residuals vs fitted values")  
plot(1:nrow(trainSet),m4$residuals, xlab="Index",  
ylab="Residuals", main="Residuals vs index")  
qqnorm(m4$residuals)  
qqline(m4$residuals, col="blue", lwd=2)



hist ( m4$residuals )  
  
# K fold cross validation to choose model selection method  
K <- 5  
validSetSplits <- sample((1:N)%%K + 1)  
RMSE1 <- c()  
RMSE2 <- c()  
RMSE3 <- c()  
RMSE4 <-c()  
for (k in 1:K) {  
 validSet <- protein1[validSetSplits==k,]  
 trainSet <- protein1[validSetSplits!=k,]   
   
 full <- lm(accuracy ~ ., data = trainSet)  
 empty <- lm(accuracy ~ 1, data = trainSet)  
   
 load("m1.rda")  
 pred1 <- predict(m1, newdata = validSet)  
 RMSE1[k] <- sqrt(mean((validSet$accuracy - pred1)^2))   
   
 load("m2.rda")  
 pred2 <- predict(m2, newdata = validSet)  
 RMSE2[k] <- sqrt(mean((validSet$accuracy - pred2)^2))   
   
 load("m3.rda")  
 pred3 <- predict(m3, newdata = validSet)  
 RMSE3[k] <- sqrt(mean((validSet$accuracy - pred3)^2))   
   
 load("m4.rda")  
 pred4 <- predict(m4, newdata = validSet)  
 RMSE4[k] <- sqrt(mean((validSet$accuracy - pred4)^2))   
  
}  
  
  
RMSE1

## [1] 0.5562111 0.5407755 0.5622529 0.5534633 0.5178010

RMSE2

## [1] 0.6539304 0.6346173 0.6130136 0.6214550 0.6191332

RMSE3

## [1] 0.5706701 0.5446036 0.5599246 0.5598758 0.5253923

RMSE4

## [1] 0.6541111 0.6276039 0.6212410 0.6393845 0.6242970

mean(RMSE1)

## [1] 0.5461008

mean(RMSE2)

## [1] 0.6284299

mean(RMSE3)

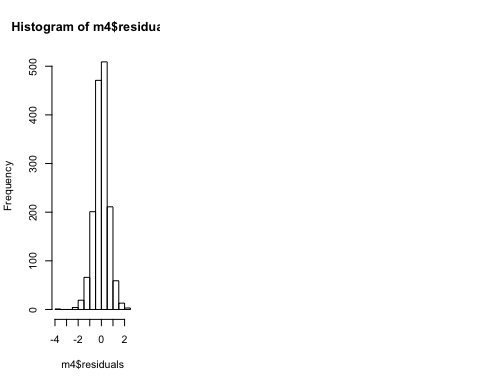
## [1] 0.5520933

mean(RMSE4)

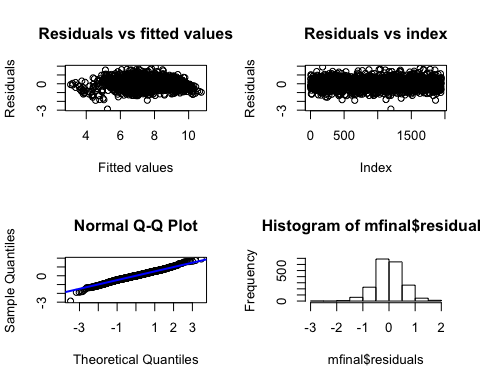
## [1] 0.6333275

# turns out m1 is indeed the better procedure among these 4 based on CV prediction error  
# if we decide on procedure m1, we can apply procedure m1 to all observations  
# to get a final model for future predictions  
# e.g.,  
full <- lm(accuracy ~ ., data = protein1)  
empty <- lm(accuracy ~ 1, data = protein1)  
  
#mfinal <- stepAIC(object = empty, scope = list(upper = full, lower = empty), direction = "forward", k = log(nrow(trainSet)))  
  
  
#save(mfinal, file= "mfinal.rda")  
load("mfinal.rda")  
  
summary(mfinal)

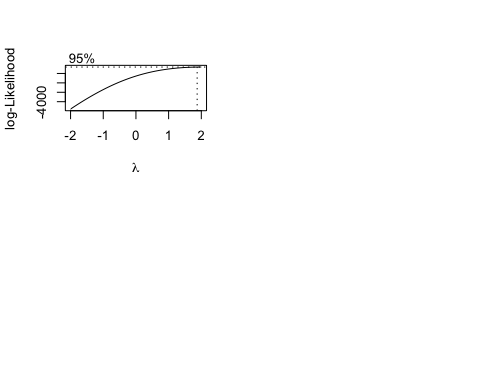
#Model Assumption Check on mfinal   
par(mfrow=c(2,2))



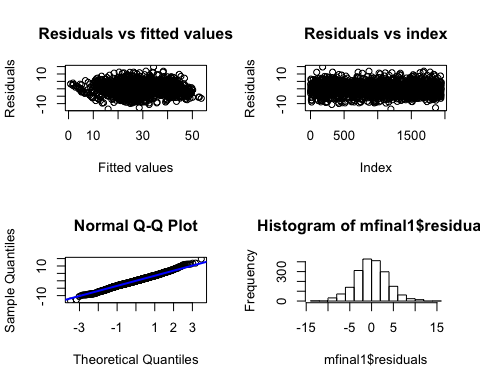
plot(mfinal$fitted.values,mfinal$residuals, xlab="Fitted values",  
ylab="Residuals", main="Residuals vs fitted values")  
plot(1:nrow(protein1),mfinal$residuals, xlab="Index",  
ylab="Residuals", main="Residuals vs index")  
qqnorm(mfinal$residuals)  
qqline(mfinal$residuals, col="blue", lwd=2)  
hist ( mfinal$residuals)



#Box-Cox Transformation   
colz <- names(mfinal$coefficients)  
colz <- colz[2:length(colz)]  
mfinal.ind <- match(colz, colnames(protein1))  
mfinal.ind <- append(1, mfinal.ind)  
  
bc <- boxcox(mfinal)  
lambda <- bc$x[which.max(bc$y)]  
mfinal1 <- lm( (accuracy^lambda - 1)/lambda ~., data=protein1[,mfinal.ind])  
  
#Model Assumption Check on mfinal1  
par(mfrow=c(2,2))



plot(mfinal1$fitted.values,mfinal1$residuals, xlab="Fitted values",  
ylab="Residuals", main="Residuals vs fitted values")  
plot(1:nrow(protein1),mfinal1$residuals, xlab="Index",  
ylab="Residuals", main="Residuals vs index")  
qqnorm(mfinal1$residuals)  
qqline(mfinal1$residuals, col="blue", lwd=2)  
hist ( mfinal1$residuals)



summary(mfinal1)

##   
## Call:  
## lm(formula = (accuracy^lambda - 1)/lambda ~ ., data = protein1[,   
## mfinal.ind])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.6683 -2.3080 -0.0632 2.3044 14.7033   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -10.55805 2.61046 -4.045 5.46e-05 \*\*\*  
## aliph1HC\_aliph2HC\_long 0.39777 0.05149 7.726 1.80e-14 \*\*\*  
## scLysN\_bbC\_vlong -0.77263 0.11453 -6.746 2.02e-11 \*\*\*  
## aliph2HC\_bbN\_medshort -0.26347 0.04040 -6.522 8.93e-11 \*\*\*  
## aliph1HC\_aliph2HC\_medlong 0.36403 0.05773 6.306 3.58e-10 \*\*\*  
## bbC\_bbC\_medshort -0.29534 0.06079 -4.858 1.29e-06 \*\*\*  
## aromaticC\_sulfur\_short -0.23550 0.10403 -2.264 0.023703 \*   
## aromaticC\_hydroxylO\_medlong -0.34871 0.04667 -7.472 1.21e-13 \*\*\*  
## aliph1HC\_aromaticC\_medshort 0.58187 0.07333 7.935 3.61e-15 \*\*\*  
## carbonylC\_aromaticC\_short 1.27262 0.19698 6.461 1.33e-10 \*\*\*  
## aliph1HC\_aromaticC\_vlong 0.61366 0.05543 11.070 < 2e-16 \*\*\*  
## bbN\_bbCA\_medlong -0.26859 0.02968 -9.050 < 2e-16 \*\*\*  
## aliph1HC\_aromaticC\_long 0.63528 0.05449 11.658 < 2e-16 \*\*\*  
## aliph1HC\_aliph1HC\_vlong 1.06365 0.16341 6.509 9.70e-11 \*\*\*  
## aliph1HC\_bbN\_vlong -0.26904 0.05148 -5.226 1.93e-07 \*\*\*  
## scAGN\_bbN\_long 0.31741 0.04079 7.781 1.18e-14 \*\*\*  
## carboxylO\_bbC\_vlong 0.30787 0.05692 5.409 7.17e-08 \*\*\*  
## aromaticC\_hydroxylO\_long -0.28274 0.03433 -8.235 3.33e-16 \*\*\*  
## bbN\_bbN\_medlong -0.24233 0.04953 -4.893 1.08e-06 \*\*\*  
## sulfur\_bbC\_medlong -0.30002 0.06748 -4.446 9.25e-06 \*\*\*  
## aliph2HC\_aromaticC\_vlong 0.11350 0.01786 6.357 2.59e-10 \*\*\*  
## aliph3HC\_bbC\_vlong 0.15585 0.02869 5.432 6.32e-08 \*\*\*  
## aliph1HC\_bbO\_long -0.28266 0.06165 -4.585 4.84e-06 \*\*\*  
## aliph1HC\_aromaticC\_medlong 0.52340 0.06053 8.647 < 2e-16 \*\*\*  
## aliph3HC\_bbN\_short 0.75269 0.12960 5.808 7.43e-09 \*\*\*  
## aliph2HC\_scArgN\_vlong 0.44922 0.15155 2.964 0.003074 \*\*   
## aliph3HC\_aromaticC\_long -0.08057 0.02604 -3.094 0.002003 \*\*   
## aromaticC\_bbCA\_vlong 0.06695 0.01880 3.561 0.000378 \*\*\*  
## bbO\_bbO\_short -0.27893 0.08031 -3.473 0.000526 \*\*\*  
## carbonylC\_aromaticC\_long 0.14859 0.04793 3.100 0.001963 \*\*   
## carboxylC\_scLysN\_vlong 1.04321 0.27383 3.810 0.000144 \*\*\*  
## sulfur\_bbC\_vlong -0.28836 0.04788 -6.022 2.07e-09 \*\*\*  
## aromaticC\_sulfur\_long 0.30625 0.06810 4.497 7.31e-06 \*\*\*  
## aliph2HC\_bbN\_medlong 0.15647 0.02452 6.382 2.20e-10 \*\*\*  
## scAGN\_carbonylO\_medshort -0.56403 0.11494 -4.907 1.00e-06 \*\*\*  
## bbO\_bbO\_long -0.11131 0.03223 -3.454 0.000565 \*\*\*  
## aliph1HC\_aliph1HC\_long 1.35104 0.22842 5.915 3.94e-09 \*\*\*  
## aliph1HC\_scArgN\_long 4.77619 0.58320 8.190 4.81e-16 \*\*\*  
## aliph1HC\_aromaticC\_short 0.92454 0.14580 6.341 2.86e-10 \*\*\*  
## carboxylC\_aromaticC\_long 0.30161 0.09989 3.019 0.002568 \*\*   
## aromaticC\_hydroxylO\_medshort -0.31168 0.06711 -4.644 3.65e-06 \*\*\*  
## carbonylC\_bbC\_medlong -0.28573 0.05994 -4.767 2.02e-06 \*\*\*  
## hydroxylO\_carbonylO\_medlong 0.31490 0.07592 4.148 3.51e-05 \*\*\*  
## aliph3HC\_hydroxylO\_long 0.34289 0.07005 4.895 1.07e-06 \*\*\*  
## bbN\_bbN\_medshort -0.64549 0.07204 -8.960 < 2e-16 \*\*\*  
## carboxylC\_bbC\_medlong 0.33229 0.12620 2.633 0.008533 \*\*   
## aliph2HC\_aromaticC\_medshort 0.13385 0.03094 4.326 1.60e-05 \*\*\*  
## scArgN\_bbO\_medlong 2.40861 0.53309 4.518 6.63e-06 \*\*\*  
## carboxylC\_carboxylC\_vlong -1.18886 0.25158 -4.726 2.47e-06 \*\*\*  
## scArgN\_carboxylO\_long -0.93417 0.15206 -6.144 9.85e-10 \*\*\*  
## carbonylC\_bbProN\_medlong -1.07902 0.19220 -5.614 2.27e-08 \*\*\*  
## scAGN\_bbCA\_medlong 0.21304 0.05615 3.794 0.000153 \*\*\*  
## aliph3HC\_aromaticC\_vlong -0.09019 0.02567 -3.513 0.000454 \*\*\*  
## carboxylO\_bbN\_vlong 0.23526 0.05682 4.141 3.62e-05 \*\*\*  
## scLysN\_carboxylO\_long 0.92755 0.18542 5.002 6.20e-07 \*\*\*  
## bbCA\_bbO\_vshort -2.47416 0.45937 -5.386 8.12e-08 \*\*\*  
## sulfur\_bbCA\_short -0.32822 0.16122 -2.036 0.041912 \*   
## carbonylC\_sulfur\_short -0.92907 0.24196 -3.840 0.000127 \*\*\*  
## bbCA\_bbC\_vlong -0.05839 0.02080 -2.807 0.005058 \*\*   
## aliph2HC\_bbN\_vlong 0.06481 0.01718 3.772 0.000167 \*\*\*  
## aliph1HC\_bbO\_vlong 0.19583 0.04663 4.200 2.80e-05 \*\*\*  
## bbProN\_carboxylO\_vlong 0.69230 0.25140 2.754 0.005949 \*\*   
## aromaticC\_scAGN\_vlong 0.13731 0.03585 3.830 0.000133 \*\*\*  
## carbonylO\_sulfur\_medshort 0.50595 0.16596 3.049 0.002331 \*\*   
## aliph1HC\_bbC\_medshort 0.38097 0.10071 3.783 0.000160 \*\*\*  
## aromaticC\_bbO\_vlong 0.06255 0.01737 3.602 0.000324 \*\*\*  
## hydroxylO\_bbC\_medlong 0.19092 0.05127 3.724 0.000202 \*\*\*  
## aromaticC\_hydroxylO\_short -0.18893 0.08366 -2.258 0.024042 \*   
## carboxylC\_bbO\_medlong 0.59767 0.12867 4.645 3.64e-06 \*\*\*  
## bbProN\_bbCA\_medshort 0.24416 0.15332 1.593 0.111427   
## carbonylC\_scAGN\_long 0.29987 0.13297 2.255 0.024237 \*   
## bbN\_bbCA\_long -0.09656 0.02590 -3.728 0.000198 \*\*\*  
## aromaticC\_bbO\_vshort -0.98885 0.25639 -3.857 0.000119 \*\*\*  
## carbonylC\_carboxylO\_short 1.40676 0.29979 4.693 2.90e-06 \*\*\*  
## carbonylC\_bbCA\_medshort 0.31115 0.08863 3.511 0.000458 \*\*\*  
## aliph1HC\_aliph3HC\_medlong -0.43459 0.10181 -4.269 2.07e-05 \*\*\*  
## scAGN\_carboxylO\_vshort -0.81688 0.33726 -2.422 0.015527 \*   
## sulfur\_sulfur\_vlong 0.58701 0.27019 2.173 0.029939 \*   
## aliph1HC\_bbProN\_medlong 1.51069 0.32447 4.656 3.45e-06 \*\*\*  
## aliph1HC\_bbN\_medshort -0.48920 0.10729 -4.559 5.46e-06 \*\*\*  
## aromaticC\_aromaticC\_vlong -0.08123 0.01617 -5.023 5.58e-07 \*\*\*  
## aliph2HC\_aliph3HC\_vlong 0.15787 0.03054 5.169 2.60e-07 \*\*\*  
## aliph3HC\_bbCA\_short 0.29993 0.15457 1.940 0.052478 .   
## aromaticC\_bbO\_medlong 0.05235 0.02754 1.901 0.057457 .   
## aliph3HC\_carboxylO\_long -0.41075 0.18742 -2.192 0.028531 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.767 on 1861 degrees of freedom  
## Multiple R-squared: 0.879, Adjusted R-squared: 0.8736   
## F-statistic: 161 on 84 and 1861 DF, p-value: < 2.2e-16

#Prediction of mfinal  
protein\_test <- read.csv('protein-test.csv')  
  
pred <- predict(mfinal1, data=protein\_test)  
writeLines(as.character(pred), "mypreds.txt")