report

ZJH

2022/3/29

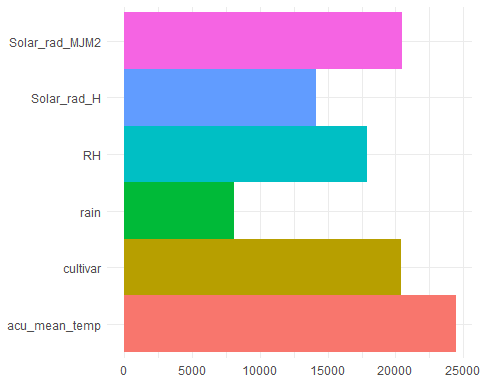
analyze\_data <- readRDS('analyze.RDS')   
numeric\_dat <- analyze\_data %>% select(.,-c(key,observe\_ys,cultivar,observe\_date:sample\_label,G1L:O9L,year,temp\_differ))  
train\_idx <- sample(1:138,108)  
test\_idx <- !(1:138 %in%train\_idx)  
train <- numeric\_dat[train\_idx,]  
test <- numeric\_dat[test\_idx,]  
  
 importanceplot <- function(pp=and\_rf) {  
 i\_scores <- varImp(pp, conditional=TRUE)  
 #Gathering rownames in 'var' and converting it to the factor  
 #to provide 'fill' parameter for the bar chart.  
 i\_scores <- i\_scores %>% tibble::rownames\_to\_column("var")  
 i\_scores$var<- i\_scores$var %>% as.factor()  
  
 #Plotting the bar and polar charts for comparing variables  
 i\_bar <- ggplot(data = i\_scores) +  
 geom\_bar(  
 stat = "identity",#it leaves the data without count and bin  
 mapping = aes(x = var, y=Overall, fill = var),  
 show.legend = FALSE,  
 width = 1  
 ) +  
 labs(x = NULL, y = NULL)  
 i\_bar + coord\_flip() + theme\_minimal()  
 }  
  
# ```  
#   
# ```{r}  
xtrain <- model.matrix(polyphenol~.+acu\_mean\_temp\*RH\*Solar\_rad\_H\*Solar\_rad\_MJM2, train)[,-1]  
ytrain <- train$polyphenol  
ytest <- test$polyphenol  
xtest <- model.matrix(polyphenol~.+acu\_mean\_temp\*RH\*Solar\_rad\_H\*Solar\_rad\_MJM2, test)[,-1] %>% as\_tibble()%>% as.matrix()  
data\_polyphenol <- analyze\_data %>% select(., c(cultivar,rain,polyphenol,acu\_mean\_temp,RH,Solar\_rad\_H,Solar\_rad\_MJM2))  
and\_rf <- randomForest(polyphenol~.+acu\_mean\_temp\*RH\*Solar\_rad\_H\*Solar\_rad\_MJM2, data = data\_polyphenol[train\_idx,])  
and\_pred <- predict(and\_rf,data\_polyphenol[test\_idx,])  
(mean((and\_pred-ytest)^2)/var(ytest))^0.5

## [1] 0.338629

accuracy(and\_pred,ytest)

## ME RMSE MAE MPE MAPE  
## Test set 2.649548 16.17038 9.504406 -1.845829 8.965136

importanceplot()



# ```  
#   
# ```{r}  
# xtrain <- model.matrix(FAA~., train)[,-1]  
# ytrain <- train$FAA  
# ytest <- test$FAA  
# xtest <- model.matrix(FAA~., test)[,-1] %>% as\_tibble() %>% select(.,season2nd:season6th,season7th,everything()) %>% as.matrix()  
# data\_FAA <- analyze\_data %>% select(., c(cultivar,FAA,rain,acu\_mean\_temp,RH,Solar\_rad\_H,Solar\_rad\_MJM2))  
# and\_rf <- randomForest(FAA~., data = data\_FAA[train\_idx,])  
# and\_pred <- predict(and\_rf,data\_FAA[test\_idx,])  
# (mean((and\_pred-ytest)^2)/var(ytest))^0.5  
# importanceplot()  
# ```  
#   
# ```{r}  
# xtrain <- model.matrix(Gallic\_Acid~., train)[,-1]  
# ytrain <- train$Gallic\_Acid  
# ytest <- test$Gallic\_Acid  
# xtest <- model.matrix(Gallic\_Acid~., test)[,-1] %>% as\_tibble() %>% select(.,season2nd:season6th,season7th,everything()) %>% as.matrix()  
# data\_Gallic\_Acid <- analyze\_data %>% select(., c(cultivar,rain,Gallic\_Acid,acu\_mean\_temp,RH,Solar\_rad\_H,Solar\_rad\_MJM2))  
# and\_rf <- randomForest(Gallic\_Acid~., data = data\_Gallic\_Acid[train\_idx,])  
# and\_pred <- predict(and\_rf,data\_Gallic\_Acid[test\_idx,])  
# (mean((and\_pred-ytest)^2)/var(ytest))^0.5  
# importanceplot()  
# ```  
# ```{r}  
# xtrain <- model.matrix(GC~., train)[,-1]  
# ytrain <- train$GC  
# ytest <- test$GC  
# xtest <- model.matrix(GC~., test)[,-1] %>% as\_tibble() %>% select(.,season2nd:season6th,season7th,everything()) %>% as.matrix()  
# data\_GC <- analyze\_data %>% select(., c(cultivar,rain,GC,acu\_mean\_temp,RH,Solar\_rad\_H,Solar\_rad\_MJM2))  
# and\_rf <- randomForest(GC~., data = data\_GC[train\_idx,])  
# and\_pred <- predict(and\_rf,data\_GC[test\_idx,])  
# (mean((and\_pred-ytest)^2)/var(ytest))^0.5  
# importanceplot()  
# ```  
# ```{r}  
# xtrain <- model.matrix(GCG~., train)[,-1]  
# ytrain <- train$GCG  
# ytest <- test$GCG  
# xtest <- model.matrix(GCG~., test)[,-1] %>% as\_tibble() %>% select(.,season2nd:season6th,season7th,everything()) %>% as.matrix()  
# data\_GCG <- analyze\_data %>% select(., c(cultivar,GCG,rain,acu\_mean\_temp,RH,Solar\_rad\_H,Solar\_rad\_MJM2))  
# and\_rf <- randomForest(GCG~., data = data\_GCG[train\_idx,])  
# and\_pred <- predict(and\_rf,data\_GCG[test\_idx,])  
# (mean((and\_pred-ytest)^2)/var(ytest))^0.5  
# importanceplot()  
# ```  
#   
# ```{r}  
# xtrain <- model.matrix(Catechin~., train)[,-1]  
# ytrain <- train$Catechin  
# ytest <- test$Catechin  
# xtest <- model.matrix(Catechin~., test)[,-1] %>% as\_tibble() %>% select(.,season2nd:season6th,season7th,everything()) %>% as.matrix()  
# data\_caffeine <- analyze\_data %>% select(., c(cultivar,Catechin,rain,acu\_mean\_temp,RH,Solar\_rad\_H,Solar\_rad\_MJM2))  
# and\_rf <- randomForest(Catechin~., data = data\_caffeine[train\_idx,])  
# and\_pred <- predict(and\_rf,data\_caffeine[test\_idx,])  
# (mean((and\_pred-ytest)^2)/var(ytest))^0.5  
# importanceplot()  
# ```  
#   
# ```{r}  
# chem <- analyze\_data %>% select(,polyphenol:total\_catechins)  
# pr\_chem <- prcomp(scale(chem) )  
# biplot(pr\_chem)  
# importanceplot()  
# ```

# {r} # xtrain <- model.matrix(level~., train)[,-1] # ytrain <- train$level # ytest <- test$level # xtest <- model.matrix(level~., test)[,-1] %>% as\_tibble() %>% select(.,season2nd:season6th,season7th,everything()) %>% as.matrix() # data\_level <- analyze\_data %>% select(., c(polyphenol:total\_catechins)) # data\_level$level <- factor(data\_level$level) # and\_rf <- randomForest(level~., data = data\_level[train\_idx,]) # and\_pred <- predict(and\_rf,data\_level[test\_idx,]) # table(data\_level[test\_idx,]$level,and\_pred) # importanceplot() #

# chem <- analyze\_data %>% select(,c(polyphenol:total\_catechins,level))  
# chem$level <- factor(chem$level)  
# chem[,1:13] <- log(chem[,1:13])  
# test\_idx <- sample(1:138,size = 30)  
# train\_idx <- not(1:138 %in% test\_idx)  
# chem\_test <- chem[test\_idx,]  
# chem\_train <- chem[train\_idx,]  
#   
# logit\_level <- glm(level~caffeine+GCG ,data=chem\_train,family=binomial())  
#logit\_level <- glm(level~Catechin+caffeine+FAA+polyphenol,data=chem,family=binomial())  
# summary(logit\_level)  
# predicted\_level <- predict(logit\_level,chem\_test)  
# predicted\_level <- ifelse(predicted\_level<=0.7,'B','C')  
# table(chem\_test$level ,predicted\_level)

cross\_validation <- function(y\_pred,ytest){  
 y\_pred <- ifelse(y\_pred<=0.7,'B','C')  
 ytest <- chem\_test$level  
 print(table(y\_pred,ytest))  
}

chem <- analyze\_data %>% select(,c(polyphenol:total\_catechins,level))  
table(chem$level)

##   
## B C   
## 51 87

chem$level <- factor(chem$level) %>% as.numeric()  
chem$level <- ifelse(chem$level>1,1,0)  
  
test\_idx <- sample(1:138,size = 30)  
train\_idx <- not(1:138 %in% test\_idx)  
chem\_test <- chem[test\_idx,]  
chem\_train <- chem[train\_idx,]  
chem\_test\_y <- chem\_test$level

# 1.建立空的線性迴歸(只有截距項)  
null <- glm(level~1 ,data=chem\_train,family=binomial())  
full <- glm(level~. ,data=chem\_train,family=binomial())  
  
# 2.使用step()，一個一個把變數丟進去  
forward.lm = step(null,   
 # 從空模型開始，一個一個丟變數，  
 # 最大不會超過完整的線性迴歸  
 # (一定要加上界 upper=full，不可以不加)   
 scope=list(lower=null, upper=full),   
 direction="forward")

## Start: AIC=142.09  
## level ~ 1  
##   
## Df Deviance AIC  
## + Theanine 1 122.22 126.22  
## + FAA 1 123.92 127.92  
## + caffeine 1 125.34 129.34  
## + Catechin 1 127.47 131.47  
## + EC 1 130.64 134.64  
## + GC 1 130.92 134.92  
## + ECG 1 132.54 136.54  
## + total\_catechins 1 134.54 138.54  
## + EGCG 1 134.88 138.88  
## <none> 140.09 142.09  
## + polyphenol 1 138.13 142.13  
## + GCG 1 138.81 142.81  
## + EGC 1 139.74 143.74  
## + Gallic\_Acid 1 140.06 144.06  
##   
## Step: AIC=126.22  
## level ~ Theanine  
##   
## Df Deviance AIC  
## + Catechin 1 108.18 114.18  
## + caffeine 1 110.77 116.77  
## + EC 1 114.61 120.61  
## + ECG 1 116.40 122.40  
## + total\_catechins 1 117.68 123.68  
## + EGCG 1 117.90 123.90  
## + GC 1 118.69 124.69  
## + GCG 1 119.32 125.32  
## + FAA 1 119.70 125.70  
## <none> 122.22 126.22  
## + EGC 1 121.46 127.46  
## + polyphenol 1 122.22 128.22  
## + Gallic\_Acid 1 122.22 128.22  
##   
## Step: AIC=114.18  
## level ~ Theanine + Catechin  
##   
## Df Deviance AIC  
## + EGC 1 102.67 110.67  
## + GC 1 105.99 113.99  
## <none> 108.18 114.18  
## + total\_catechins 1 106.91 114.91  
## + polyphenol 1 107.16 115.16  
## + caffeine 1 107.16 115.16  
## + FAA 1 107.31 115.31  
## + EGCG 1 107.73 115.73  
## + ECG 1 107.93 115.93  
## + Gallic\_Acid 1 108.09 116.09  
## + GCG 1 108.13 116.13  
## + EC 1 108.15 116.15  
##   
## Step: AIC=110.67  
## level ~ Theanine + Catechin + EGC  
##   
## Df Deviance AIC  
## + caffeine 1 96.736 106.74  
## + EC 1 99.995 110.00  
## <none> 102.669 110.67  
## + EGCG 1 101.316 111.32  
## + total\_catechins 1 101.419 111.42  
## + FAA 1 102.053 112.05  
## + ECG 1 102.220 112.22  
## + GCG 1 102.392 112.39  
## + Gallic\_Acid 1 102.606 112.61  
## + GC 1 102.659 112.66  
## + polyphenol 1 102.666 112.67  
##   
## Step: AIC=106.74  
## level ~ Theanine + Catechin + EGC + caffeine  
##   
## Df Deviance AIC  
## + GC 1 94.364 106.36  
## <none> 96.736 106.74  
## + GCG 1 94.805 106.81  
## + ECG 1 95.236 107.24  
## + FAA 1 95.827 107.83  
## + total\_catechins 1 96.050 108.05  
## + EC 1 96.174 108.17  
## + polyphenol 1 96.425 108.42  
## + Gallic\_Acid 1 96.443 108.44  
## + EGCG 1 96.567 108.57  
##   
## Step: AIC=106.36  
## level ~ Theanine + Catechin + EGC + caffeine + GC  
##   
## Df Deviance AIC  
## <none> 94.364 106.36  
## + GCG 1 92.588 106.59  
## + FAA 1 93.504 107.50  
## + ECG 1 93.793 107.79  
## + EC 1 93.972 107.97  
## + Gallic\_Acid 1 94.271 108.27  
## + polyphenol 1 94.310 108.31  
## + EGCG 1 94.335 108.33  
## + total\_catechins 1 94.353 108.35

backward.lm = step(full,   
 # 這裡可以加下界(lower=null)，也可以不加  
 scope = list(upper=full),   
 direction="backward")

## Start: AIC=118.82  
## level ~ polyphenol + FAA + Theanine + caffeine + Gallic\_Acid +   
## Catechin + GC + EGC + EC + EGCG + GCG + ECG + total\_catechins  
##   
## Df Deviance AIC  
## - Catechin 1 90.833 116.83  
## - EC 1 90.835 116.83  
## - EGCG 1 90.835 116.83  
## - total\_catechins 1 90.836 116.84  
## - GCG 1 90.836 116.84  
## - EGC 1 90.836 116.84  
## - GC 1 90.836 116.84  
## - ECG 1 90.836 116.84  
## - FAA 1 90.838 116.84  
## - Gallic\_Acid 1 90.847 116.85  
## - polyphenol 1 91.045 117.05  
## <none> 90.822 118.82  
## - caffeine 1 95.063 121.06  
## - Theanine 1 99.025 125.03  
##   
## Step: AIC=116.83  
## level ~ polyphenol + FAA + Theanine + caffeine + Gallic\_Acid +   
## GC + EGC + EC + EGCG + GCG + ECG + total\_catechins  
##   
## Df Deviance AIC  
## - FAA 1 90.848 114.85  
## - Gallic\_Acid 1 90.861 114.86  
## - polyphenol 1 91.066 115.07  
## <none> 90.833 116.83  
## - caffeine 1 95.114 119.11  
## - EC 1 96.834 120.83  
## - ECG 1 98.393 122.39  
## - EGCG 1 98.981 122.98  
## - Theanine 1 99.171 123.17  
## - total\_catechins 1 99.292 123.29  
## - GCG 1 99.661 123.66  
## - GC 1 99.848 123.85  
## - EGC 1 100.739 124.74  
##   
## Step: AIC=114.85  
## level ~ polyphenol + Theanine + caffeine + Gallic\_Acid + GC +   
## EGC + EC + EGCG + GCG + ECG + total\_catechins  
##   
## Df Deviance AIC  
## - Gallic\_Acid 1 90.884 112.88  
## - polyphenol 1 91.090 113.09  
## <none> 90.848 114.85  
## - caffeine 1 95.592 117.59  
## - EC 1 96.863 118.86  
## - ECG 1 98.477 120.48  
## - EGCG 1 98.994 120.99  
## - total\_catechins 1 99.320 121.32  
## - GC 1 99.889 121.89  
## - GCG 1 100.150 122.15  
## - EGC 1 100.807 122.81  
## - Theanine 1 102.169 124.17  
##   
## Step: AIC=112.88  
## level ~ polyphenol + Theanine + caffeine + GC + EGC + EC + EGCG +   
## GCG + ECG + total\_catechins  
##   
## Df Deviance AIC  
## - polyphenol 1 91.097 111.10  
## <none> 90.884 112.88  
## - caffeine 1 96.781 116.78  
## - EC 1 96.903 116.90  
## - EGCG 1 99.079 119.08  
## - ECG 1 99.245 119.25  
## - total\_catechins 1 99.453 119.45  
## - GC 1 99.922 119.92  
## - GCG 1 100.223 120.22  
## - EGC 1 101.045 121.05  
## - Theanine 1 103.363 123.36  
##   
## Step: AIC=111.1  
## level ~ Theanine + caffeine + GC + EGC + EC + EGCG + GCG + ECG +   
## total\_catechins  
##   
## Df Deviance AIC  
## <none> 91.097 111.10  
## - EC 1 96.903 114.90  
## - caffeine 1 96.911 114.91  
## - EGCG 1 99.252 117.25  
## - total\_catechins 1 99.632 117.63  
## - ECG 1 99.680 117.68  
## - GCG 1 100.250 118.25  
## - GC 1 100.337 118.34  
## - EGC 1 101.384 119.38  
## - Theanine 1 103.875 121.88

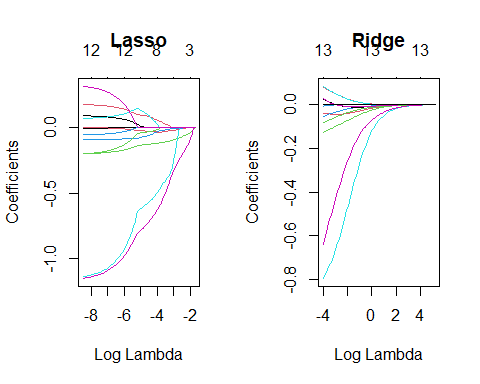
y\_pred\_forward <- predict(forward.lm,chem\_test)  
cross\_validation(y\_pred\_forward,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 10 9  
## C 3 8

y\_pred\_backward <- predict(backward.lm,chem\_test)  
cross\_validation(y\_pred\_backward,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 11 7  
## C 2 10

# lasso & ridge for chem ----  
ridge = glmnet(x = as.matrix(chem[, -14]),   
 y = chem[, 14],   
 alpha = 0,  
 family = "binomial")  
  
lasso = glmnet(x = as.matrix(chem[, -14]),   
 y = chem[, 14],   
 alpha = 1,  
 family = "binomial")  
  
par(mfcol = c(1, 2)) # cool!!  
plot(lasso, xvar='lambda', main="Lasso")  
plot(ridge, xvar='lambda', main="Ridge")



# 經由 cv 的手法，評估每個模型在不同 lambda 下   
# 的 cvm(mean cross-validated error)  
cv.lasso = cv.glmnet(x = as.matrix(chem[, -14]),   
 y = chem[, 14],   
 alpha = 1, # lasso  
 family = "binomial")  
  
# 評估每個模型的 cvm(mean cross-validated error)後  
# 取最小 cvm 模型所對應的 lambda  
best.lambda\_lasso = cv.lasso$lambda.min  
  
  
plot(lasso, xvar='lambda', main="Lasso")  
abline(v=log(best.lambda\_lasso), col="blue", lty=5.5 )  
# 觀察哪些變數被挑選出來，其係數不為 0的那些  
coef(cv.lasso, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 2.93673531  
## polyphenol .   
## FAA -0.02729770  
## Theanine -0.12945097  
## caffeine -0.06515799  
## Gallic\_Acid -0.58626020  
## Catechin -0.74067470  
## GC .   
## EGC 0.08716775  
## EC -0.03638274  
## EGCG .   
## GCG 0.10662670  
## ECG .   
## total\_catechins .

select.ind = which(coef(cv.lasso, s = "lambda.min") != 0)  
select.ind = select.ind[-1]-1 # remove `Intercept` and 平移剩下的ind  
select.ind # 第幾個變數是重要的 (不看 `Intercept`)

## [1] 2 3 4 5 6 8 9 11

select.varialbes = colnames(chem)[select.ind]  
select.varialbes

## [1] "FAA" "Theanine" "caffeine" "Gallic\_Acid" "Catechin"   
## [6] "EGC" "EC" "GCG"

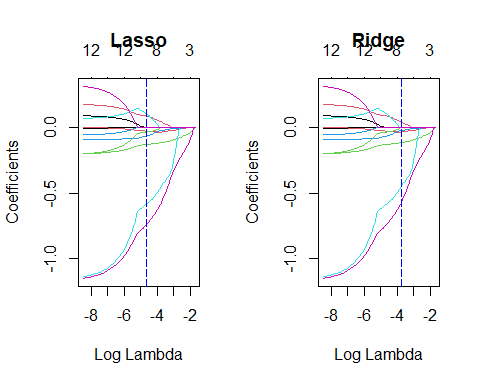
lasso.lm <- glm(level ~ ., family = 'binomial',data = chem\_test[, c(select.varialbes, "level")])  
lasso.test <- predict(lasso.lm,chem\_test)  
cross\_validation(lasso.test,ytest)

## ytest  
## y\_pred 0 1  
## B 12 4  
## C 1 13

# 經由 cv 的手法，評估每個模型在不同 lambda 下   
# 的 cvm(mean cross-validated error)  
cv.ridge = cv.glmnet(x = as.matrix(chem[, -14]),   
 y = chem[, 14],   
 alpha = 0, # lasso  
 family = "binomial")  
  
# 評估每個模型的 cvm(mean cross-validated error)後  
# 取最小 cvm 模型所對應的 lambda  
best.lambda = cv.ridge$lambda.min  
best.lambda

## [1] 0.02415429

plot(lasso, xvar='lambda', main="Ridge")  
abline(v=log(best.lambda), col="blue", lty=5.5 )



# 觀察哪些變數被挑選出來，其係數不為 0的那些  
coef(cv.lasso, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 2.93673531  
## polyphenol .   
## FAA -0.02729770  
## Theanine -0.12945097  
## caffeine -0.06515799  
## Gallic\_Acid -0.58626020  
## Catechin -0.74067470  
## GC .   
## EGC 0.08716775  
## EC -0.03638274  
## EGCG .   
## GCG 0.10662670  
## ECG .   
## total\_catechins .

select.ind = which(coef(cv.lasso, s = "lambda.min") != 0)  
select.ind = select.ind[-1]-1 # remove `Intercept` and 平移剩下的ind  
select.ind # 第幾個變數是重要的 (不看 `Intercept`)

## [1] 2 3 4 5 6 8 9 11

select.varialbes = colnames(chem)[select.ind]  
select.varialbes

## [1] "FAA" "Theanine" "caffeine" "Gallic\_Acid" "Catechin"   
## [6] "EGC" "EC" "GCG"

ridge.lm <- glm(level ~ ., family = 'binomial',data = chem\_train[, c(select.varialbes, "level")])  
ridge.test <- predict(ridge.lm,chem\_test)  
cross\_validation(ridge.test,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 11 6  
## C 2 11

chem <- analyze\_data %>% select(,c(polyphenol:total\_catechins,level))  
chem$level <- factor(chem$level) %>% as.numeric()  
chem$level <- ifelse(chem$level>1,1,0)  
  
chem[,1:13] <- log(chem[,1:13])  
chem <- chem[,-6]  
test\_idx <- sample(1:138,size = 30)  
train\_idx <- not(1:138 %in% test\_idx)  
chem\_test <- chem[test\_idx,]  
chem\_train <- chem[train\_idx,]  
chem\_test\_y <- chem\_test$level

# 1.建立空的線性迴歸(只有截距項)  
null <- glm(level~1 ,data=chem\_train,family=binomial())  
full <- glm(level~. ,data=chem\_train,family=binomial())  
  
# 2.使用step()，一個一個把變數丟進去  
forward.lm = step(null,   
 # 從空模型開始，一個一個丟變數，  
 # 最大不會超過完整的線性迴歸  
 # (一定要加上界 upper=full，不可以不加)   
 scope=list(lower=null, upper=full),   
 direction="forward")

## Start: AIC=138.06  
## level ~ 1  
##   
## Df Deviance AIC  
## + caffeine 1 118.08 122.08  
## + FAA 1 118.79 122.79  
## + EGCG 1 120.72 124.72  
## + total\_catechins 1 121.98 125.98  
## + ECG 1 124.60 128.60  
## + Theanine 1 125.04 129.04  
## + GC 1 126.23 130.23  
## + EC 1 126.81 130.81  
## + polyphenol 1 126.89 130.89  
## + EGC 1 131.84 135.84  
## <none> 136.06 138.06  
## + GCG 1 135.52 139.52  
## + Gallic\_Acid 1 135.83 139.83  
##   
## Step: AIC=122.07  
## level ~ caffeine  
##   
## Df Deviance AIC  
## + GCG 1 96.638 102.64  
## + FAA 1 103.078 109.08  
## + Theanine 1 108.549 114.55  
## <none> 118.075 122.08  
## + polyphenol 1 116.230 122.23  
## + Gallic\_Acid 1 116.565 122.56  
## + EGCG 1 117.348 123.35  
## + EGC 1 117.429 123.43  
## + GC 1 117.500 123.50  
## + EC 1 117.593 123.59  
## + ECG 1 117.937 123.94  
## + total\_catechins 1 118.020 124.02  
##   
## Step: AIC=102.64  
## level ~ caffeine + GCG  
##   
## Df Deviance AIC  
## + FAA 1 91.456 99.456  
## + EGCG 1 92.259 100.259  
## + Theanine 1 92.975 100.975  
## + Gallic\_Acid 1 93.378 101.378  
## <none> 96.638 102.638  
## + EGC 1 94.879 102.879  
## + EC 1 95.236 103.236  
## + ECG 1 95.817 103.817  
## + total\_catechins 1 95.872 103.872  
## + polyphenol 1 96.429 104.429  
## + GC 1 96.620 104.620  
##   
## Step: AIC=99.46  
## level ~ caffeine + GCG + FAA  
##   
## Df Deviance AIC  
## + Gallic\_Acid 1 87.737 97.737  
## + EGCG 1 89.368 99.368  
## <none> 91.456 99.456  
## + EC 1 89.515 99.515  
## + EGC 1 89.679 99.679  
## + Theanine 1 89.778 99.778  
## + ECG 1 91.284 101.284  
## + GC 1 91.366 101.366  
## + total\_catechins 1 91.391 101.391  
## + polyphenol 1 91.449 101.449  
##   
## Step: AIC=97.74  
## level ~ caffeine + GCG + FAA + Gallic\_Acid  
##   
## Df Deviance AIC  
## + EC 1 78.888 90.888  
## + EGCG 1 84.376 96.376  
## + EGC 1 85.440 97.440  
## <none> 87.737 97.737  
## + Theanine 1 86.844 98.844  
## + ECG 1 87.120 99.120  
## + polyphenol 1 87.340 99.340  
## + GC 1 87.528 99.528  
## + total\_catechins 1 87.684 99.684  
##   
## Step: AIC=90.89  
## level ~ caffeine + GCG + FAA + Gallic\_Acid + EC  
##   
## Df Deviance AIC  
## + ECG 1 72.163 86.163  
## + EGCG 1 73.321 87.321  
## + total\_catechins 1 74.263 88.263  
## <none> 78.888 90.888  
## + polyphenol 1 77.264 91.264  
## + Theanine 1 78.248 92.248  
## + GC 1 78.758 92.758  
## + EGC 1 78.857 92.857  
##   
## Step: AIC=86.16  
## level ~ caffeine + GCG + FAA + Gallic\_Acid + EC + ECG  
##   
## Df Deviance AIC  
## <none> 72.163 86.163  
## + GC 1 70.662 86.662  
## + EGCG 1 71.539 87.539  
## + total\_catechins 1 71.621 87.621  
## + EGC 1 71.839 87.839  
## + Theanine 1 72.026 88.026  
## + polyphenol 1 72.114 88.114

backward.lm = step(full,   
 # 這裡可以加下界(lower=null)，也可以不加  
 scope = list(upper=full),   
 direction="backward")

## Start: AIC=92.04  
## level ~ polyphenol + FAA + Theanine + caffeine + Gallic\_Acid +   
## GC + EGC + EC + EGCG + GCG + ECG + total\_catechins  
##   
## Df Deviance AIC  
## - ECG 1 66.078 90.078  
## - Theanine 1 66.825 90.825  
## - polyphenol 1 66.854 90.854  
## - EGC 1 67.610 91.610  
## - EGCG 1 67.880 91.880  
## <none> 66.038 92.038  
## - total\_catechins 1 68.464 92.464  
## - FAA 1 69.788 93.788  
## - GC 1 70.714 94.714  
## - EC 1 77.507 101.507  
## - Gallic\_Acid 1 78.711 102.711  
## - caffeine 1 82.370 106.370  
## - GCG 1 92.790 116.790  
##   
## Step: AIC=90.08  
## level ~ polyphenol + FAA + Theanine + caffeine + Gallic\_Acid +   
## GC + EGC + EC + EGCG + GCG + total\_catechins  
##   
## Df Deviance AIC  
## - polyphenol 1 66.889 88.889  
## - Theanine 1 67.028 89.028  
## <none> 66.078 90.078  
## - EGCG 1 68.950 90.950  
## - FAA 1 69.856 91.856  
## - EGC 1 70.159 92.159  
## - total\_catechins 1 70.958 92.958  
## - GC 1 71.828 93.828  
## - EC 1 77.526 99.526  
## - Gallic\_Acid 1 80.093 102.093  
## - caffeine 1 83.878 105.878  
## - GCG 1 93.429 115.429  
##   
## Step: AIC=88.89  
## level ~ FAA + Theanine + caffeine + Gallic\_Acid + GC + EGC +   
## EC + EGCG + GCG + total\_catechins  
##   
## Df Deviance AIC  
## - Theanine 1 67.619 87.619  
## <none> 66.889 88.889  
## - EGCG 1 69.215 89.215  
## - EGC 1 70.208 90.208  
## - FAA 1 70.502 90.502  
## - total\_catechins 1 70.973 90.973  
## - GC 1 71.933 91.933  
## - EC 1 77.562 97.562  
## - Gallic\_Acid 1 82.560 102.560  
## - caffeine 1 86.645 106.645  
## - GCG 1 93.766 113.766  
##   
## Step: AIC=87.62  
## level ~ FAA + caffeine + Gallic\_Acid + GC + EGC + EC + EGCG +   
## GCG + total\_catechins  
##   
## Df Deviance AIC  
## <none> 67.619 87.619  
## - EGCG 1 70.793 88.793  
## - EGC 1 72.001 90.001  
## - FAA 1 72.403 90.403  
## - total\_catechins 1 72.723 90.723  
## - GC 1 72.741 90.741  
## - EC 1 79.232 97.232  
## - Gallic\_Acid 1 84.612 102.612  
## - caffeine 1 87.285 105.285  
## - GCG 1 95.463 113.463

print(forward.lm)

##   
## Call: glm(formula = level ~ caffeine + GCG + FAA + Gallic\_Acid + EC +   
## ECG, family = binomial(), data = chem\_train)  
##   
## Coefficients:  
## (Intercept) caffeine GCG FAA Gallic\_Acid EC   
## 34.793 -11.202 8.194 -2.558 4.002 3.718   
## ECG   
## -4.739   
##   
## Degrees of Freedom: 107 Total (i.e. Null); 101 Residual  
## Null Deviance: 136.1   
## Residual Deviance: 72.16 AIC: 86.16

summary(forward.lm)

##   
## Call:  
## glm(formula = level ~ caffeine + GCG + FAA + Gallic\_Acid + EC +   
## ECG, family = binomial(), data = chem\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2778 -0.3285 0.1662 0.4799 2.2630   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 34.793 7.861 4.426 9.59e-06 \*\*\*  
## caffeine -11.202 2.768 -4.047 5.19e-05 \*\*\*  
## GCG 8.194 2.067 3.964 7.36e-05 \*\*\*  
## FAA -2.558 1.338 -1.912 0.055924 .   
## Gallic\_Acid 4.002 1.084 3.690 0.000224 \*\*\*  
## EC 3.718 1.077 3.452 0.000556 \*\*\*  
## ECG -4.739 2.014 -2.353 0.018620 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 136.059 on 107 degrees of freedom  
## Residual deviance: 72.163 on 101 degrees of freedom  
## AIC: 86.163  
##   
## Number of Fisher Scoring iterations: 6

print(backward.lm)

##   
## Call: glm(formula = level ~ FAA + caffeine + Gallic\_Acid + GC + EGC +   
## EC + EGCG + GCG + total\_catechins, family = binomial(), data = chem\_train)  
##   
## Coefficients:  
## (Intercept) FAA caffeine Gallic\_Acid   
## 101.953 -3.215 -10.967 4.517   
## GC EGC EC EGCG   
## 2.014 8.499 6.489 17.242   
## GCG total\_catechins   
## 10.665 -39.446   
##   
## Degrees of Freedom: 107 Total (i.e. Null); 98 Residual  
## Null Deviance: 136.1   
## Residual Deviance: 67.62 AIC: 87.62

summary(backward.lm)

##   
## Call:  
## glm(formula = level ~ FAA + caffeine + Gallic\_Acid + GC + EGC +   
## EC + EGCG + GCG + total\_catechins, family = binomial(), data = chem\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2263 -0.3288 0.1131 0.4484 2.5067   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 101.9534 34.2966 2.973 0.002952 \*\*   
## FAA -3.2152 1.6895 -1.903 0.057035 .   
## caffeine -10.9667 3.1634 -3.467 0.000527 \*\*\*  
## Gallic\_Acid 4.5169 1.2956 3.486 0.000490 \*\*\*  
## GC 2.0136 0.9393 2.144 0.032060 \*   
## EGC 8.4990 4.3190 1.968 0.049089 \*   
## EC 6.4892 2.2454 2.890 0.003853 \*\*   
## EGCG 17.2421 10.1926 1.692 0.090717 .   
## GCG 10.6647 2.7267 3.911 9.18e-05 \*\*\*  
## total\_catechins -39.4461 19.0636 -2.069 0.038529 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 136.059 on 107 degrees of freedom  
## Residual deviance: 67.619 on 98 degrees of freedom  
## AIC: 87.619  
##   
## Number of Fisher Scoring iterations: 7

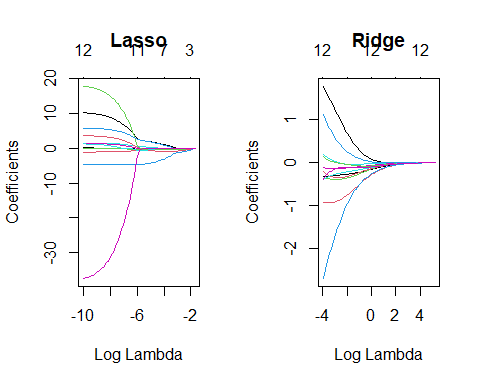
y\_pred\_forward <- predict(forward.lm,chem\_test)  
cross\_validation(y\_pred\_forward,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 12 6  
## C 4 8

y\_pred\_backward <- predict(backward.lm,chem\_test)  
cross\_validation(y\_pred\_backward,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 12 4  
## C 4 10

# lasso & ridge for chem ----  
ridge = glmnet(x = as.matrix(chem[, -13]),   
 y = chem[, 13],   
 alpha = 0,  
 family = "binomial")  
  
lasso = glmnet(x = as.matrix(chem[, -13]),   
 y = chem[, 13],   
 alpha = 1,  
 family = "binomial")  
  
par(mfcol = c(1, 2)) # cool!!  
plot(lasso, xvar='lambda', main="Lasso")  
plot(ridge, xvar='lambda', main="Ridge")



# 經由 cv 的手法，評估每個模型在不同 lambda 下   
# 的 cvm(mean cross-validated error)  
cv.lasso = cv.glmnet(x = as.matrix(chem[, -13]),   
 y = chem[, 13],   
 alpha = 1, # lasso  
 family = "binomial")  
  
# 評估每個模型的 cvm(mean cross-validated error)後  
# 取最小 cvm 模型所對應的 lambda  
best.lambda\_lasso = cv.lasso$lambda.min  
  
  
plot(lasso, xvar='lambda', main="Lasso")  
abline(v=log(best.lambda\_lasso), col="blue", lty=5.5 )  
# 觀察哪些變數被挑選出來，其係數不為 0的那些  
coef(cv.lasso, s = "lambda.min")

## 13 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 8.42354625  
## polyphenol .   
## FAA -0.86783771  
## Theanine -0.39561258  
## caffeine -2.77469064  
## Gallic\_Acid .   
## GC .   
## EGC 0.92283752  
## EC -0.01088735  
## EGCG .   
## GCG 0.73054757  
## ECG .   
## total\_catechins .

select.ind = which(coef(cv.lasso, s = "lambda.min") != 0)  
select.ind = select.ind[-1]-1 # remove `Intercept` and 平移剩下的ind  
select.ind # 第幾個變數是重要的 (不看 `Intercept`)

## [1] 2 3 4 7 8 10

select.varialbes = colnames(chem)[select.ind]  
select.varialbes

## [1] "FAA" "Theanine" "caffeine" "EGC" "EC" "GCG"

lasso.lm <- glm(level ~ ., family = 'binomial',data = chem\_train[, c(select.varialbes, "level")])  
print(lasso.lm)

##   
## Call: glm(formula = level ~ ., family = "binomial", data = chem\_train[,   
## c(select.varialbes, "level")])  
##   
## Coefficients:  
## (Intercept) FAA Theanine caffeine EGC EC   
## 22.3941 -1.8785 -0.4982 -7.9004 0.5840 0.7626   
## GCG   
## 3.9115   
##   
## Degrees of Freedom: 107 Total (i.e. Null); 101 Residual  
## Null Deviance: 136.1   
## Residual Deviance: 87.38 AIC: 101.4

summary(lasso.lm)

##   
## Call:  
## glm(formula = level ~ ., family = "binomial", data = chem\_train[,   
## c(select.varialbes, "level")])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4752 -0.4980 0.2678 0.6889 2.1691   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 22.3941 5.4793 4.087 4.37e-05 \*\*\*  
## FAA -1.8785 1.0634 -1.767 0.07730 .   
## Theanine -0.4982 0.3954 -1.260 0.20766   
## caffeine -7.9004 2.0111 -3.928 8.55e-05 \*\*\*  
## EGC 0.5840 1.2719 0.459 0.64611   
## EC 0.7626 0.7637 0.999 0.31800   
## GCG 3.9115 1.3008 3.007 0.00264 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 136.059 on 107 degrees of freedom  
## Residual deviance: 87.376 on 101 degrees of freedom  
## AIC: 101.38  
##   
## Number of Fisher Scoring iterations: 6

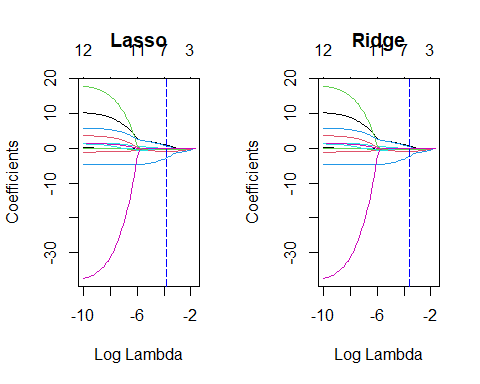
lasso.test <- predict(lasso.lm,chem\_test)  
cross\_validation(lasso.test,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 11 6  
## C 5 8

# 經由 cv 的手法，評估每個模型在不同 lambda 下   
# 的 cvm(mean cross-validated error)  
cv.ridge = cv.glmnet(x = as.matrix(chem[, -13]),   
 y = chem[, 13],   
 alpha = 0, # lasso  
 family = "binomial")  
  
# 評估每個模型的 cvm(mean cross-validated error)後  
# 取最小 cvm 模型所對應的 lambda  
best.lambda = cv.ridge$lambda.min  
best.lambda

## [1] 0.02735718

plot(lasso, xvar='lambda', main="Ridge")  
abline(v=log(best.lambda), col="blue", lty=5.5 )



# 觀察哪些變數被挑選出來，其係數不為 0的那些  
coef(cv.lasso, s = "lambda.min")

## 13 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 8.42354625  
## polyphenol .   
## FAA -0.86783771  
## Theanine -0.39561258  
## caffeine -2.77469064  
## Gallic\_Acid .   
## GC .   
## EGC 0.92283752  
## EC -0.01088735  
## EGCG .   
## GCG 0.73054757  
## ECG .   
## total\_catechins .

select.ind = which(coef(cv.lasso, s = "lambda.min") != 0)  
select.ind = select.ind[-1]-1 # remove `Intercept` and 平移剩下的ind  
select.ind # 第幾個變數是重要的 (不看 `Intercept`)

## [1] 2 3 4 7 8 10

select.varialbes = colnames(chem)[select.ind]  
select.varialbes

## [1] "FAA" "Theanine" "caffeine" "EGC" "EC" "GCG"

ridge.lm <- glm(level ~ ., family = 'binomial',data = chem\_train[, c(select.varialbes, "level")])  
print(ridge.lm)

##   
## Call: glm(formula = level ~ ., family = "binomial", data = chem\_train[,   
## c(select.varialbes, "level")])  
##   
## Coefficients:  
## (Intercept) FAA Theanine caffeine EGC EC   
## 22.3941 -1.8785 -0.4982 -7.9004 0.5840 0.7626   
## GCG   
## 3.9115   
##   
## Degrees of Freedom: 107 Total (i.e. Null); 101 Residual  
## Null Deviance: 136.1   
## Residual Deviance: 87.38 AIC: 101.4

summary(ridge.lm)

##   
## Call:  
## glm(formula = level ~ ., family = "binomial", data = chem\_train[,   
## c(select.varialbes, "level")])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4752 -0.4980 0.2678 0.6889 2.1691   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 22.3941 5.4793 4.087 4.37e-05 \*\*\*  
## FAA -1.8785 1.0634 -1.767 0.07730 .   
## Theanine -0.4982 0.3954 -1.260 0.20766   
## caffeine -7.9004 2.0111 -3.928 8.55e-05 \*\*\*  
## EGC 0.5840 1.2719 0.459 0.64611   
## EC 0.7626 0.7637 0.999 0.31800   
## GCG 3.9115 1.3008 3.007 0.00264 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 136.059 on 107 degrees of freedom  
## Residual deviance: 87.376 on 101 degrees of freedom  
## AIC: 101.38  
##   
## Number of Fisher Scoring iterations: 6

ridge.test <- predict(ridge.lm,chem\_test)  
cross\_validation(ridge.test,chem\_test\_y)

## ytest  
## y\_pred 0 1  
## B 11 6  
## C 5 8

scatter\_hist\_plot <- function(dat) {  
 n = length(colnames(dat))  
 for (i in 1:n)   
 {  
 for(j in 1:n)  
 {  
 if(i==j)  
 {  
 test <- ggplot(data=dat,aes(get(colnames(dat[i]))))+  
 geom\_histogram(bins = 30,fill = 'darkgreen')+  
 labs(x=paste(i,colnames(dat[i]),y=NULL))  
 }  
 else  
 {  
 test <- ggplot(data=dat,aes(get(colnames(dat[i])),get(colnames(dat[j]))),colour='green')+  
 geom\_point()+  
 labs(x=paste(i,colnames(dat[i])),y=paste(j,colnames(dat[j])))  
 }  
 print(test,vp=viewport(width=1/n,   
 height=1/n,  
 x=((i-1)%%n)/n+1/(2\*n),  
 y=((j-1)%%n)/n+1/(2\*n)  
 )  
 )  
 }  
 }  
}

weather <- analyze\_data %>% select(.,c(polyphenol,acu\_mean\_temp:Solar\_rad\_MJM2))  
  
# png('var\_scatter\_hist\_weather\_chem.png',width = 3400,height=3400)  
# scatter\_hist\_plot(weather\_chem)  
# dev.off()

test\_idx <- sample(1:138,size = 30)  
train\_idx <- not(1:138 %in% test\_idx)  
weather\_test <- weather[test\_idx,]  
weather\_train <- weather[train\_idx,]  
weather\_test\_y <- weather\_test$polyphenol

# 1.建立空的線性迴歸(只有截距項)  
null <- lm(polyphenol~1 ,data=weather\_train)  
full <- lm(polyphenol~. ,data=weather\_train)  
  
# 2.使用step()，一個一個把變數丟進去  
forward.lm = step(null,   
 # 從空模型開始，一個一個丟變數，  
 # 最大不會超過完整的線性迴歸  
 # (一定要加上界 upper=full，不可以不加)   
 scope=list(lower=null, upper=full),   
 direction="forward")

## Start: AIC=776.4  
## polyphenol ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + Solar\_rad\_MJM2 1 31537.2 108897 750.93  
## + acu\_mean\_temp 1 19832.7 120601 761.96  
## + Solar\_rad\_H 1 11443.5 128990 769.22  
## + temp\_differ 1 8371.8 132062 771.76  
## <none> 140434 776.40  
## + RH 1 1386.4 139047 777.33  
## + rain 1 424.8 140009 778.07  
## + growth\_mean\_temp 1 1.6 140432 778.40  
##   
## Step: AIC=750.93  
## polyphenol ~ Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## + temp\_differ 1 5765.9 103131 747.05  
## <none> 108897 750.93  
## + acu\_mean\_temp 1 1086.5 107810 751.85  
## + RH 1 93.6 108803 752.84  
## + Solar\_rad\_H 1 15.2 108881 752.92  
## + rain 1 9.7 108887 752.92  
## + growth\_mean\_temp 1 0.8 108896 752.93  
##   
## Step: AIC=747.05  
## polyphenol ~ Solar\_rad\_MJM2 + temp\_differ  
##   
## Df Sum of Sq RSS AIC  
## <none> 103131 747.05  
## + acu\_mean\_temp 1 968.99 102162 748.04  
## + Solar\_rad\_H 1 668.63 102462 748.35  
## + growth\_mean\_temp 1 261.59 102869 748.78  
## + rain 1 165.17 102965 748.88  
## + RH 1 18.22 103112 749.04

backward.lm = step(full,   
 # 這裡可以加下界(lower=null)，也可以不加  
 scope = list(upper=full),   
 direction="backward")

## Start: AIC=753.33  
## polyphenol ~ acu\_mean\_temp + rain + growth\_mean\_temp + temp\_differ +   
## RH + Solar\_rad\_H + Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## - rain 1 147.8 99779 751.49  
## - Solar\_rad\_H 1 510.1 100142 751.88  
## - growth\_mean\_temp 1 688.6 100320 752.07  
## - RH 1 1407.4 101039 752.84  
## <none> 99631 753.33  
## - acu\_mean\_temp 1 2528.0 102160 754.03  
## - temp\_differ 1 3977.5 103609 755.55  
## - Solar\_rad\_MJM2 1 4489.0 104121 756.09  
##   
## Step: AIC=751.49  
## polyphenol ~ acu\_mean\_temp + growth\_mean\_temp + temp\_differ +   
## RH + Solar\_rad\_H + Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## - Solar\_rad\_H 1 373.0 100152 749.89  
## - growth\_mean\_temp 1 756.6 100536 750.30  
## - RH 1 1342.6 101122 750.93  
## <none> 99779 751.49  
## - acu\_mean\_temp 1 2381.0 102160 752.03  
## - temp\_differ 1 4151.7 103931 753.89  
## - Solar\_rad\_MJM2 1 4369.6 104149 754.12  
##   
## Step: AIC=749.89  
## polyphenol ~ acu\_mean\_temp + growth\_mean\_temp + temp\_differ +   
## RH + Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## - growth\_mean\_temp 1 438.4 100591 748.36  
## - RH 1 1023.1 101175 748.99  
## <none> 100152 749.89  
## - acu\_mean\_temp 1 2527.9 102680 750.58  
## - Solar\_rad\_MJM2 1 5441.3 105594 753.60  
## - temp\_differ 1 6567.3 106720 754.75  
##   
## Step: AIC=748.36  
## polyphenol ~ acu\_mean\_temp + temp\_differ + RH + Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## - RH 1 1570.9 102162 748.04  
## <none> 100591 748.36  
## - acu\_mean\_temp 1 2521.6 103112 749.04  
## - Solar\_rad\_MJM2 1 5092.9 105684 751.70  
## - temp\_differ 1 6164.4 106755 752.79  
##   
## Step: AIC=748.04  
## polyphenol ~ acu\_mean\_temp + temp\_differ + Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## - acu\_mean\_temp 1 969.0 103131 747.05  
## <none> 102162 748.04  
## - temp\_differ 1 5648.3 107810 751.85  
## - Solar\_rad\_MJM2 1 17836.2 119998 763.41  
##   
## Step: AIC=747.05  
## polyphenol ~ temp\_differ + Solar\_rad\_MJM2  
##   
## Df Sum of Sq RSS AIC  
## <none> 103131 747.05  
## - temp\_differ 1 5765.9 108897 750.93  
## - Solar\_rad\_MJM2 1 28931.3 132062 771.76

summary(forward.lm)

##   
## Call:  
## lm(formula = polyphenol ~ Solar\_rad\_MJM2 + temp\_differ, data = weather\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -57.295 -22.413 -4.882 15.188 108.663   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.26524 16.01779 0.516 0.6069   
## Solar\_rad\_MJM2 0.18629 0.03432 5.427 3.7e-07 \*\*\*  
## temp\_differ -0.08880 0.03665 -2.423 0.0171 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 31.34 on 105 degrees of freedom  
## Multiple R-squared: 0.2656, Adjusted R-squared: 0.2516   
## F-statistic: 18.99 on 2 and 105 DF, p-value: 9.133e-08

summary(backward.lm)

##   
## Call:  
## lm(formula = polyphenol ~ temp\_differ + Solar\_rad\_MJM2, data = weather\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -57.295 -22.413 -4.882 15.188 108.663   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.26524 16.01779 0.516 0.6069   
## temp\_differ -0.08880 0.03665 -2.423 0.0171 \*   
## Solar\_rad\_MJM2 0.18629 0.03432 5.427 3.7e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 31.34 on 105 degrees of freedom  
## Multiple R-squared: 0.2656, Adjusted R-squared: 0.2516   
## F-statistic: 18.99 on 2 and 105 DF, p-value: 9.133e-08

y\_pred\_forward <- predict(forward.lm,weather\_test)  
accuracy(y\_pred\_forward,weather\_test\_y)

## ME RMSE MAE MPE MAPE  
## Test set -1.510633 28.13228 21.74006 -10.16212 25.16793

y\_pred\_backward <- predict(backward.lm,weather\_test)  
accuracy(y\_pred\_backward,weather\_test\_y)

## ME RMSE MAE MPE MAPE  
## Test set -1.510633 28.13228 21.74006 -10.16212 25.16793