```r
library(caTools)
library(tidyverse)
library(miscTools)
library(Metrics)
library(plotly)
library(glmnet)
library(PRROC)
install.packages("ROCit")
library(ROCit)


data = read.csv("/Users/bennetthellman/Desktop/OneDrive - Massachusetts
Institute of Technology/AE/HWs/HW2/framingham.csv")
data$TenYearCHD <- factor(data$TenYearCHD)
data$male <- factor(data$male)
data$currentSmoker <- factor(data$currentSmoker)
data$BPMeds <- factor(data$BPMeds)
data$prevalentStroke <- factor(data$prevalentStroke)
data$prevalentHyp <- factor(data$prevalentHyp)
data$diabetes <- factor(data$diabetes)
set.seed(38)
N <- nrow(data)
idx = sample.split(data$TenYearCHD, 0.75)
train <- data[idx,]
test = data[!idx,]


##############################
#a
ggplot(data, aes(sysBP, after_stat(count), fill = TenYearCHD)) +
  geom_bar(position = "fill", width = 5)+
  xlab("Systolic Blood Pressure") + labs(fill='Chronic Heart Disease') +
  theme(legend.text=element_text(size=12),
        axis.title=element_text(size=14))

ggplot(data, aes(diaBP, after_stat(count), fill = TenYearCHD)) +
  geom_bar(position = "fill", width = 5)+
  xlab("Diastolic Blood Pressure") + labs(fill='Chronic Heart Disease') +
  theme(legend.text=element_text(size=12),
        axis.title=element_text(size=14))

##############################
#b
lgm<-glm(TenYearCHD ~ ., data = data, family = "binomial")
summary(lgm)


##############################
#c
x.train = model.matrix(TenYearCHD ~ . - 1  ,
                       data=train)
y.train = train$TenYearCHD # Here, we are only including the dependent
variable.
x.test = model.matrix(TenYearCHD ~ . - 1,
                      data=test)
y.test = test$TenYearCHD
lambdas.lasso <- exp(seq(10, -10, -.01))


#five fold
system.time(cv.lasso.five <- cv.glmnet(x.train,
                      y.train,alpha=1,
                      lambda=lambdas.lasso,
                      nfolds=5, type.measure = "deviance",
family="binomial"))
plot(cv.lasso.five)
cv.lasso.five$lambda.min
coefficients(cv.lasso.five)

#tenfold
system.time(cv.lasso.ten <- cv.glmnet(x.train,
                                  y.train,alpha=1,
                                  lambda=lambdas.lasso,
                                  nfolds=10, type.measure = "deviance",
family="binomial"))
plot(cv.lasso.ten)
cv.lasso.ten$lambda.min
coefficients(cv.lasso.ten)

#leave out one CV
system.time(cv.lasso.lv <- cv.glmnet(x.train,
                                  y.train,alpha=1,
                                  lambda=lambdas.lasso,
                                  nfolds=nrow(x.train), type.measure =
"deviance", family="binomial"))
plot(cv.lasso.lv)
cv.lasso.lv$lambda.min
coefficients(cv.lasso.lv)


##############################
#d
#di
alpha = seq(0,1,.01)

lgm_pred = predict(lgm, test, type = "response")
lasso_five_pred = predict(cv.lasso.five, x.test, type = "response")
lasso_ten_pred = predict(cv.lasso.ten, x.test, type = "response")
lasso_lv_pred = predict(cv.lasso.lv, x.test, type = "response")

lgm_p = c()
five_p = c()
ten_p = c()
lv_p = c()

for (i in alpha){
  count = sum(lgm_pred > i)
  lgm_p <- c(lgm_p , count)
```

```
}

for (i in alpha){
  count = sum(lasso_ten_pred > i)
  ten_p <- c(ten_p , count)
}


baseline = rep(0,length(alpha))
ideal = length(which(y.test==1))
ideal = rep(ideal, length(alpha))

count_df = data.frame("Alpha"= alpha,"Logistic_Regression"=
lgm_p,"Ten_Fold_CV_Lasso"= ten_p,"Ideal"= ideal, "Baseline" = baseline)
ggplot(count_df, aes(x=Alpha)) +
  geom_line(aes(y = Logistic_Regression, color = "Logistic"), size = 1.5) +
  geom_line(aes(y = Ten_Fold_CV_Lasso, color = "10-Fold CV Lasso"), size=1.
5) +
  geom_line(aes(y = Baseline, color = "Baseline"),  size=1.5) +
  geom_line(aes(y = Ideal, color = "Ideal"), size=1.5) + labs(x = "alpha",
y = "# Patients Treated",color = "Legend") + scale_color_manual(values =
c("Orange", "Red", "Green", "Blue" ))

#dii
threshold_all <- seq(0, 1, .01)
profit_lgm <- c()
for (thresh in threshold_all){
  treated <- lgm_pred >= thresh
  CHD <- test$TenYearCHD == 1
  earnings <- 165000 * sum(CHD & !treated) + 165000/2.3 * sum(CHD &
treated)  + 7500 * sum(treated)
  profit_lgm <- c(profit_lgm, earnings)
}

profit_tf <- c()
for (thresh in threshold_all){
  treated2 <- lasso_ten_pred >= thresh
  CHD <- test$TenYearCHD == 1
  earnings2 <- 165000 * sum(CHD & !treated2) + 165000/2.3 * sum(CHD &
treated2) + 7500 * sum(treated2)
  profit_tf <- c(profit_tf, earnings2)
}

# Let us compute the baseline profit for comparison purposes (baseline:
treat everybody)
baseline.none.profit = 165000 * sum(test$TenYearCHD == 1)
baseline.all.profit = 165000/2.3 * sum(test$TenYearCHD == 1) + 7500 *
sum(test$TenYearCHD == 0) + 7500 * sum(test$TenYearCHD == 1)
ideal.profit = 165000/2.3 * sum(test$TenYearCHD == 1) + 7500 *
sum(test$TenYearCHD == 1)

# We record everything in a new data frame
profit_threshold <- data.frame(threshold=threshold_all,
                               logisticprofit=profit_lgm,
                               lassoprofit=profit_tf,
                               baseline_all=baseline.all.profit,
                               baseline_none = baseline.none.profit,
                               ideal=ideal.profit)


profit_threshold %>%
  ggplot(aes(x=threshold)) +
  geom_line(aes(y = logisticprofit, color = "Logistic"), size = 1.5) +
  geom_line(aes(y = lassoprofit, color = "10-Fold CV Lasso"), size=1.5) +
  geom_line(aes(y = baseline_all, color = "Baseline - Treat Everybody"),
size=1.5) +
  geom_line(aes(y = baseline_none, color = "Baseline - Treat Nobody"),
size=1.5) +
  geom_line(aes(y = ideal, color = "Ideal"), size=1.5) + labs(x = "alpha",
y = "$ Cost",color = "Legend") + scale_color_manual(values = c("Orange",
"Red", "Green", "Blue", "Purple" ))





#diii
rocr.pred.lgm <- prediction(lgm_pred, test$TenYearCHD)
perf.lgm <- performance(rocr.pred.lgm, "tpr", "fpr")
rocr.pred.df.lgm <- data.frame(fpr=slot(perf.lgm, "x.values")[[1]],
                           tpr=slot(perf.lgm, "y.values")[[1]])

rocr.pred.tf <- prediction(ten_cv_pred, test$TenYearCHD)
perf.tf <- performance(rocr.pred.tf, "tpr", "fpr")
rocr.pred.df.tf<- data.frame(fpr=slot(perf.tf, "x.values")[[1]],
                               tpr=slot(perf.tf, "y.values")[[1]])

rocr.pred.bl <- prediction(rep(0, length(test$TenYearCHD)),
test$TenYearCHD)
perf.bl <- performance(rocr.pred.bl, "tpr", "fpr")
rocr.pred.df.bl<- data.frame(fpr=slot(perf.bl, "x.values")[[1]],
                             tpr=slot(perf.bl, "y.values")[[1]])

#Ananya Krishnan showed me how to do this
df_lay = data.frame("Ideal" = as.numeric(y.test)-1)
rocr.pred.id <- prediction(df_lay$Ideal, test$TenYearCHD)
perf.id <- performance(rocr.pred.id, "tpr", "fpr")
rocr.pred.df.id<- data.frame(fpr=slot(perf.id, "x.values")[[1]],
                             tpr=slot(perf.id, "y.values")[[1]])



ggplot() +
  geom_line(data = rocr.pred.df.lgm, aes(x=fpr, y=tpr, color = "Logistic"),
size = 1.5) +
  geom_line(data = rocr.pred.df.tf, aes(x=fpr, y=tpr, color = "10-Fold CV
Lasso"), size=1.5) +
  geom_line(data = rocr.pred.df.bl, aes(x=fpr, y=tpr, color = "Baseline"),
size=1.5) +
  geom_line(data = rocr.pred.df.id, aes(x=fpr, y=tpr, color = "Ideal"),
size=1.5) + labs(x = "False Positive Rate", y = "True Positive Rate",color
= "Legend") + scale_color_manual(values = c("Orange", "Red", "Green",
"Blue" ))


#iv
lgm_auc <- performance(rocr.pred.lgm ,"auc")@y.values[[1]]
lgm_auc
tf_auc <- performance(rocr.pred.tf ,"auc")@y.values[[1]]
tf_auc
bl_auc <- performance(rocr.pred.bl ,"auc")@y.values[[1]]
bl_auc
id_auc <- performance(rocr.pred.id ,"auc")@y.values[[1]]
id_auc
```