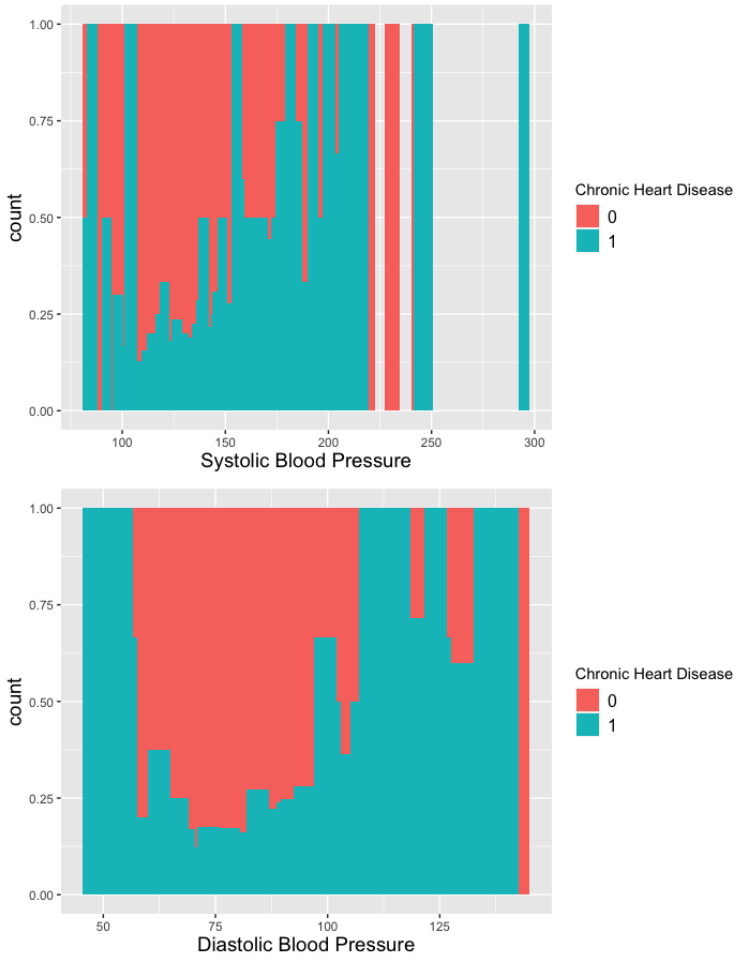
Short HW #2

Bennett Hellman
15.072 - Advanced Analytics Edge
MIT

October 8, 2021

Figure 1: Conditional Density Plots



1. (a) i.

   ii. When looking at the the middle ranges of blood pressures, larger values are associated with more occurrences of chronic heart disease. Additionally, both the lowest and highest values for each blood pressure is associated with a relative increase in proportion of those with chronic heart disease.

(b) i. The raw coefficients are the associated effect of a one-unit increase in a specific variable in terms of a change in log odds of having chronic heart disease. In order to calculate the change in probability, one would need to calculate $\Delta p_i = \frac{\exp(\beta_i \times \Delta x_i))}{1+\exp(\beta_i \times \Delta x_i)}$. The one unit increase interpretation is simple for continuous variables. For indicator variables, such as male, the estimate is associated with being in that category (e.g. being a male).

Figure 2: Saturated Logit Model Coefficients

```
Coefficients:
                                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                              -8.324218   0.706554 -11.781  < 2e-16 ***
male1                                     0.534965   0.109901   4.868 1.13e-06 ***
age                                       0.062229   0.006756   9.211  < 2e-16 ***
educationHigh school/GED                 -0.132187   0.177604  -0.744  0.45671
educationSome college/vocational school -0.136310   0.197512  -0.690  0.49011
educationSome high school                 0.059625   0.164620   0.362  0.71720
currentSmoker1                            0.073036   0.156749   0.466  0.64126
cigsPerDay                                0.018005   0.006234   2.888  0.00387 **
BPMeds1                                   0.165206   0.234484   0.705  0.48109
prevalentStroke1                          0.704867   0.491479   1.434  0.15152
prevalentHyp1                             0.233424   0.138202   1.689  0.09122 .
diabetes1                                 0.025920   0.316132   0.082  0.93465
totChol                                   0.002377   0.001129   2.105  0.03527 *
sysBP                                     0.015456   0.003812   4.054 5.03e-05 ***
diaBP                                    -0.004121   0.006444  -0.640  0.52247
BMI                                       0.005215   0.012786   0.408  0.68338
heartRate                                -0.003004   0.004213  -0.713  0.47592
glucose                                   0.007216   0.002234   3.229  0.00124 **
```

(c) i. 5 Fold Cross Validation:
Computational Time = 3.629 s
Outputs: Figure 3
Selected $\lambda = 0.0074$
Coefficients: Figure 4

ii. 10 Fold Cross Validation:
Computational Time = 5.366 s
Outputs: Figure 5
Selected $\lambda = 0.0033$
Coefficients: Figure 6

iii. Leave-out-one Cross Validation:
Computational Time = 858.748 s
Outputs: Figure 7
Selected $\lambda = 0.0048$
Coefficients: Figure 8

Figure 3: 5 Fold CV λ

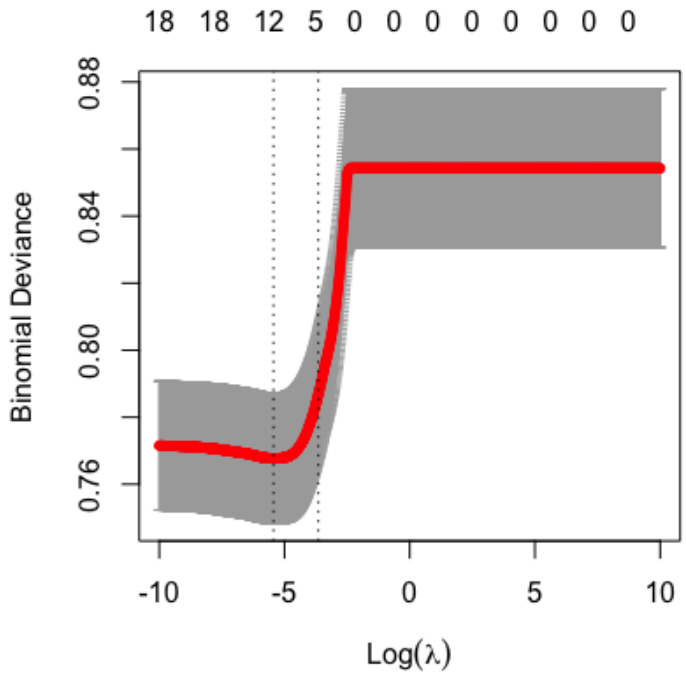


Figure 4: 5 Fold CV Coefficients

(Intercept)	-3.444853023
male0	.
male1	.
age	0.022844842
educationHigh school/GED	.
educationSome college/vocational school	.
educationSome high school	.
currentSmoker1	.
cigsPerDay	.
BPMeds1	.
prevalentStroke1	.
prevalentHyp1	.
diabetes1	.
totChol	.
sysBP	0.004360719
diaBP	.
BMI	.
heartRate	.
glucose	.

Figure 5: 10 Fold CV λ

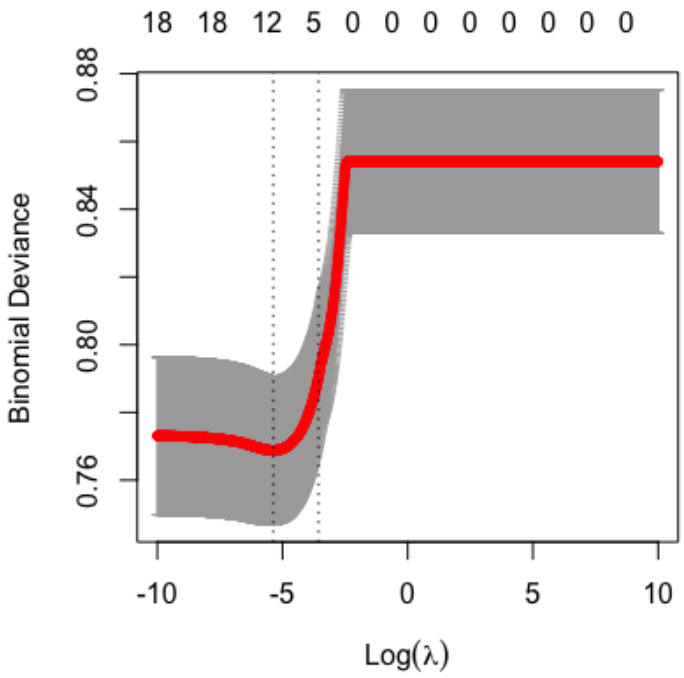


Figure 6: 10 Fold CV Coefficients

(Intercept)	-5.056235e+00
male0	-5.832977e-02
male1	8.346919e-16
age	3.980607e-02
educationHigh school/GED	.
educationSome college/vocational school	.
educationSome high school	.
currentSmoker1	.
cigsPerDay	.
BPMeds1	.
prevalentStroke1	.
prevalentHyp1	3.701117e-02
diabetes1	.
totChol	.
sysBP	9.190603e-03
diaBP	.
BMI	.
heartRate	.
glucose	1.211505e-03

Figure 7: Leave-Out-One CV λ

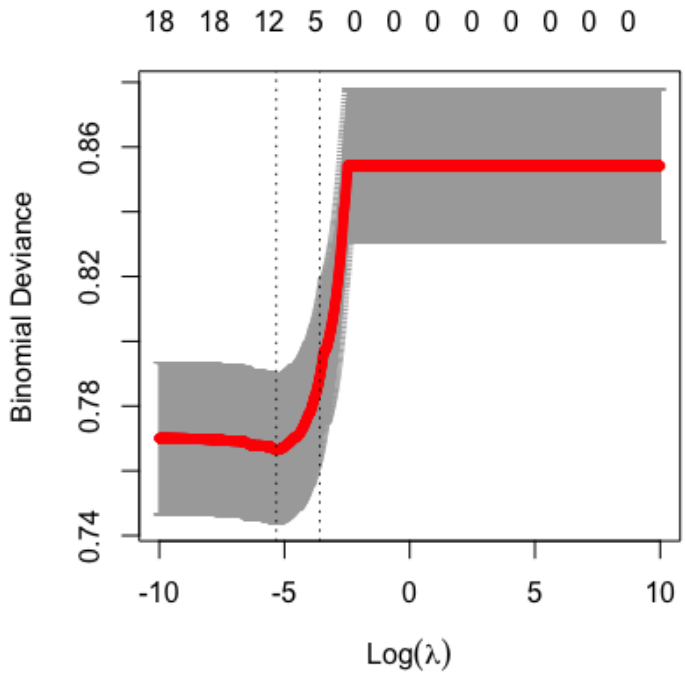
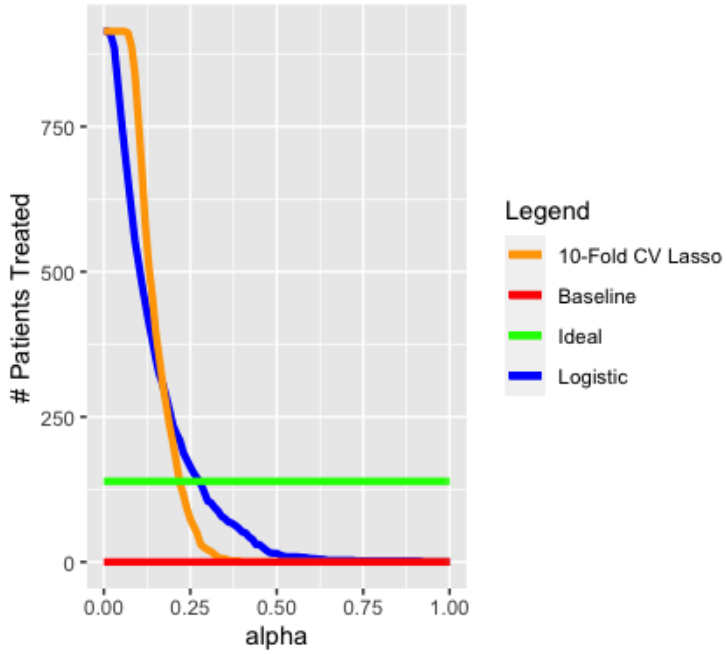


Figure 8: Leave Out One CV Coefficients

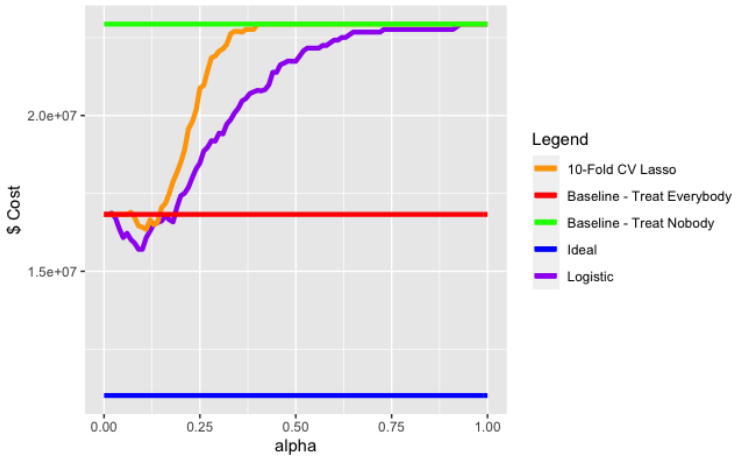
(Intercept)	-5.180587326
male0	-0.092590157
male1	.
age	0.041157927
educationHigh school/GED	.
educationSome college/vocational school	.
educationSome high school	.
currentSmoker1	.
cigsPerDay	.
BPMeds1	.
prevalentStroke1	.
prevalentHyp1	0.053178011
diabetes1	.
totChol	.
sysBP	0.009414093
diaBP	.
BMI	.
heartRate	.
glucose	0.001636845

Figure 9: Patients Treated by CHD Probability Threshold



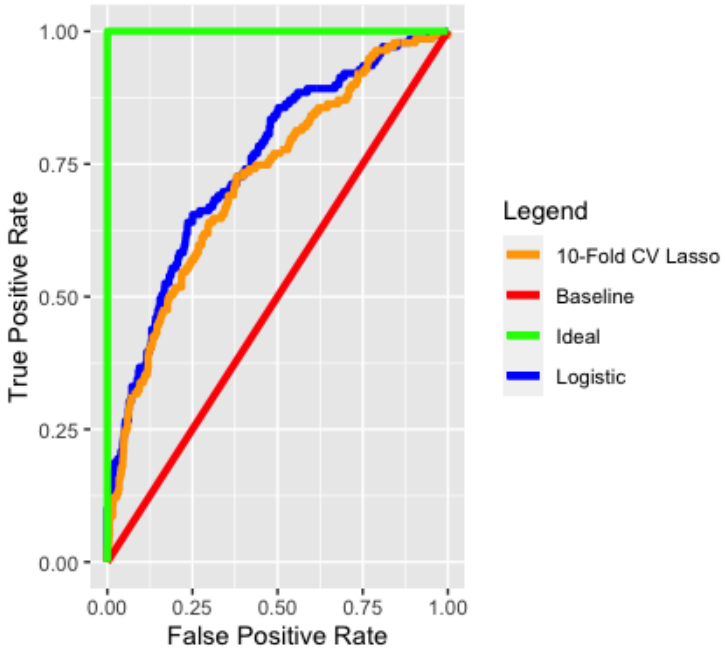
(d) i.

Figure 10: Model's Cost Curve by CHD Probability Threshold



ii.

Figure 11: Model ROC Curves



iii.

iv. Logistic Regression AUC = 0.75
10-Fold CV AUC = 0.72
Baseline AUC = 0.5
Ideal AUC = 1

v. For Figure 9, our 10-fold cross-validated lasso model predicts the most patients needing to be treated for lower thresholds but eventually predicts less than a logistic regression. Both models predict the correct number, but not necessarily the correct patients, around $\alpha = .25$.

Figure 10 demonstrates that both our lasso and logistic models cost more than they ideally should. However, for lower thresholds they cost less than treating everybody and for most thresholds cost less than treating nobody. For every threshold, lasso outperforms logistic regression in terms of expected cost.

Conversely, logistic regression proves to be superior when examining the ROC curve and AUC.

Based on these figures, no single model proves to be definitively superior. Recommendations should be made based on the stakeholder's preferences and priorities in terms of cost, patients treated, and accuracy.

```
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
```