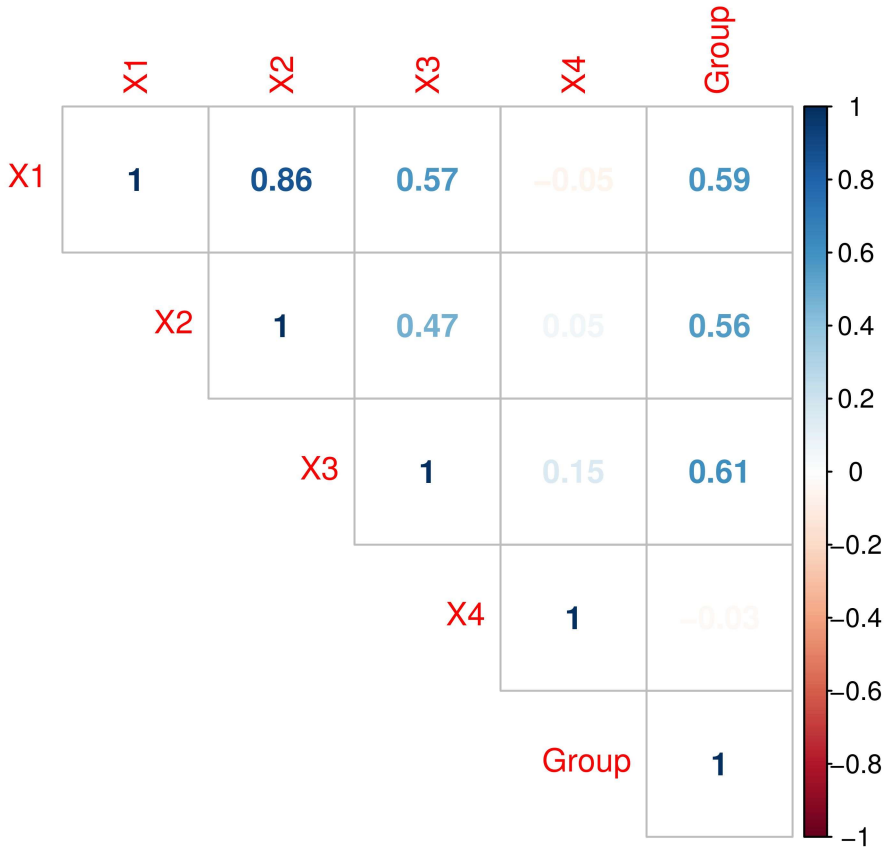



```
# it appears that all variables but x4 would be good to predict group with
```

2(b)

We test full and null model alongside some variable selection model to determine a good model to predict bankruptcy.

```
set.seed(1)
# question 2 part b ----
attach(bankruptcy)
```

```
## The following object is masked from admission:
##
##     Group
```

```
fit1 <- glm(Group ~., family = binomial, data = bankruptcy)
summary(fit1)
```

```
##
## Call:
## glm(formula = Group ~ ., family = binomial, data = bankruptcy)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.30416  -0.44545   0.00725   0.49102   2.62396
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.320      2.366  -2.248  0.02459 *
## X1             7.138      6.002   1.189  0.23433
## X2            -3.703     13.670  -0.271  0.78647
## X3             3.415      1.204   2.837  0.00455 **
## X4            -2.968      3.065  -0.968  0.33286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 63.421  on 45  degrees of freedom
## Residual deviance: 27.443  on 41  degrees of freedom
## AIC: 37.443
##
## Number of Fisher Scoring iterations: 7
```

```
# par(mfrow = c(2,2))
# plot(fit1) # to be used if we want to verify model assumptions
fit2 = glm(Group ~X1+X3, family = binomial, data = bankruptcy)
summary(fit2)
```

```
##
## Call:
## glm(formula = Group ~ X1 + X3, family = binomial, data = bankruptcy)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.26853  -0.47678   0.00942   0.48365   2.70538
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.940      1.985  -2.992  0.00277 **
## X1             6.556      2.905   2.257  0.02402 *
## X3             3.019      1.002   3.013  0.00259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 63.421  on 45  degrees of freedom
## Residual deviance: 28.636  on 43  degrees of freedom
## AIC: 34.636
##
## Number of Fisher Scoring iterations: 6
```

```
fit3 = glm(Group ~ 1, family = binomial, data = bankruptcy)
summary(fit3)
```

```
##
## Call:
## glm(formula = Group ~ 1, family = binomial, data = bankruptcy)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.252  -1.252   1.104   1.104   1.104
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.1744     0.2960   0.589    0.556
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 63.421  on 45  degrees of freedom
## Residual deviance: 63.421  on 45  degrees of freedom
## AIC: 65.421
##
## Number of Fisher Scoring iterations: 3
```

```
# compare fit 2 and 1
anova(fit2, fit1, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Group ~ X1 + X3
## Model 2: Group ~ X1 + X2 + X3 + X4
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        43     28.636
## 2        41     27.443  2   1.1924   0.5509
```

```
# since we accept the H0 we keep the reduced model fit2
# compare fit 2 with fit 3 (null model)
anova(fit2, fit3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Group ~ X1 + X3
## Model 2: Group ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        43     28.636
## 2        45     63.421 -2  -34.786 2.795e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# since we reject the H0 we keep the full model fit2
# Since we are using all of the data as training data, there is no need to split

# It seems cash flow and current assets over current
# liabilities determine bankruptcy the best.
# This can be seen as does one have enough compensation for their given risk.
```

Question 3(a)

Our goal is to use the previous question's logistic regression model for a confusion matrix, sensitivity, specificity, misclassification rate, and ROC curve.
Noted in the comments, we decided to split data 50/50 for train/test in order to achieve meaningful numbers. If tested and trained on the full data, the numbers are more or less the same for each model.

```
set.seed(1)
# question 3 part a ----
# if we proceed with our training data == test data then our following
# outcomes on each model will be the same (tested)
# almost as if it were copied and pasted
# so we will split the data down the middle, 50/50 training/testing.
# There is no motivation for using 50/50, simply using for fun

# split data 50/50, train/test
set.seed(1)
n = nrow(bankruptcy)
sampler = sample(1:n, n/2) # n/2 is the 50/50 splitter
train = bankruptcy[sampler, ]
test = bankruptcy[-sampler, ]

# create model
fit = glm(Group ~X1+X3, family = binomial, data = train)

# Estimated probabilities for test data
prob <- predict(fit, test, type = "response")

# Predicted classes (using 0.5 cutoff)
pred <- ifelse(prob >= 0.5, "nonbankrupt", "bankrupt")

# Test error rate
1 - mean(pred == test[, "Group"])
```

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

```
# Confusion matrix and (sensitivity, specificity)
# `+' = nonbankrupt, `-' = bankrupt
con.mat = table(pred, test[, "Group"])
con.mat
```

```
##
## pred           0  1
##   bankrupt    10  2
##   nonbankrupt  1 10
```

```
#       PRED CLASS
# TRUE     TN FP
# CLASS    FN TP
```

```r
# Sensitivity, TP/P = 0.8333333
10/(10+2)
```

```
## [1] 0.8333333
```

```r
con.mat[4]/sum(con.mat[3],con.mat[4])
```

```
## [1] 0.8333333
```

```r
# Specificity, TN/N = 0.9090909
10/(10+1)
```

```
## [1] 0.9090909
```

```r
con.mat[1]/sum(con.mat[1],con.mat[2])
```

```
## [1] 0.9090909
```

```r
# Overall misclassification, (FN+FP)/(N+P) = 0.1304348
# or (1-sens)*[P/(P+N)]+(1-spec)*[N/(P+N)] = 0.1304348
(1+2)/(11+12)
```

```
## [1] 0.1304348
```

```r
sum(con.mat[2],con.mat[3])/sum(con.mat)
```

```
## [1] 0.1304348
```

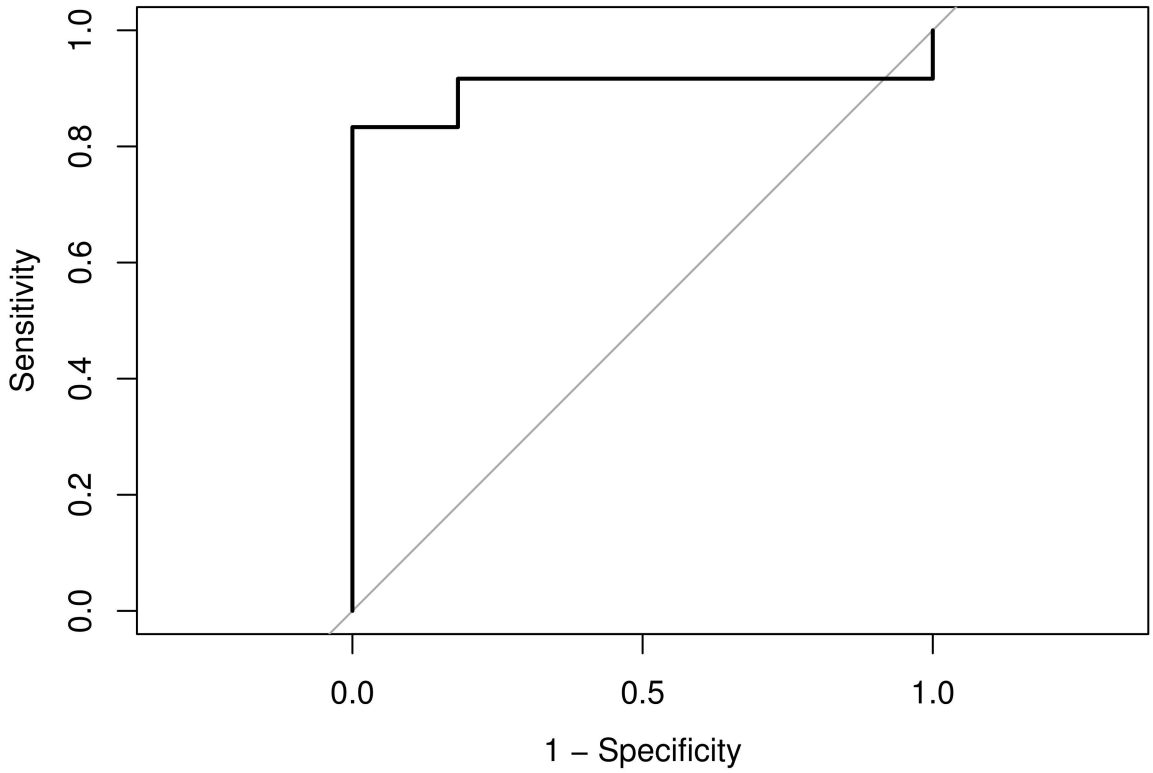
```r
# Misclassification is not bad, we will see how it ranks against the next models

# ROC curve
# case = '+' (or nonbankrupt, 1) , control = '-' (or bankrupt, 0)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
roc <- roc(test[, "Group"], prob, levels = c(0, 1))
plot(roc, legacy.axes = T)
```




```
lr_red = c("LR reduced", sum(con.mat[2],con.mat[3])/sum(con.mat), roc$auc)
```

3(b)

We perform the same analysis as 3(a), but now with all predictors.
We note in comments about comparison.

```
set.seed(1)
# question 3 part b ----
# create model with all predictors
fit = glm(Group ~., family = binomial, data = train)

# Estimated probabilities for test data
prob <- predict(fit, test, type = "response")

# Predicted classes (using 0.5 cutoff)
pred <- ifelse(prob >= 0.5, "nonbankrupt", "bankrupt")

# Test error rate
1 - mean(pred == test[, "Group"])
```

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

```r
# Confusion matrix and (sensitivity, specificity)
# `+' = nonbankrupt, `-' = bankrupt
con.mat = table(pred, test[, "Group"])
con.mat
```

```
##
## pred           0  1
##   bankrupt     9  2
##   nonbankrupt  2 10
```

```r
#       PRED CLASS
# TRUE     TN FP
# CLASS    FN TP
# Sensitivity, TP/P = 0.8333333
con.mat[4]/sum(con.mat[3],con.mat[4])
```

```
## [1] 0.8333333
```

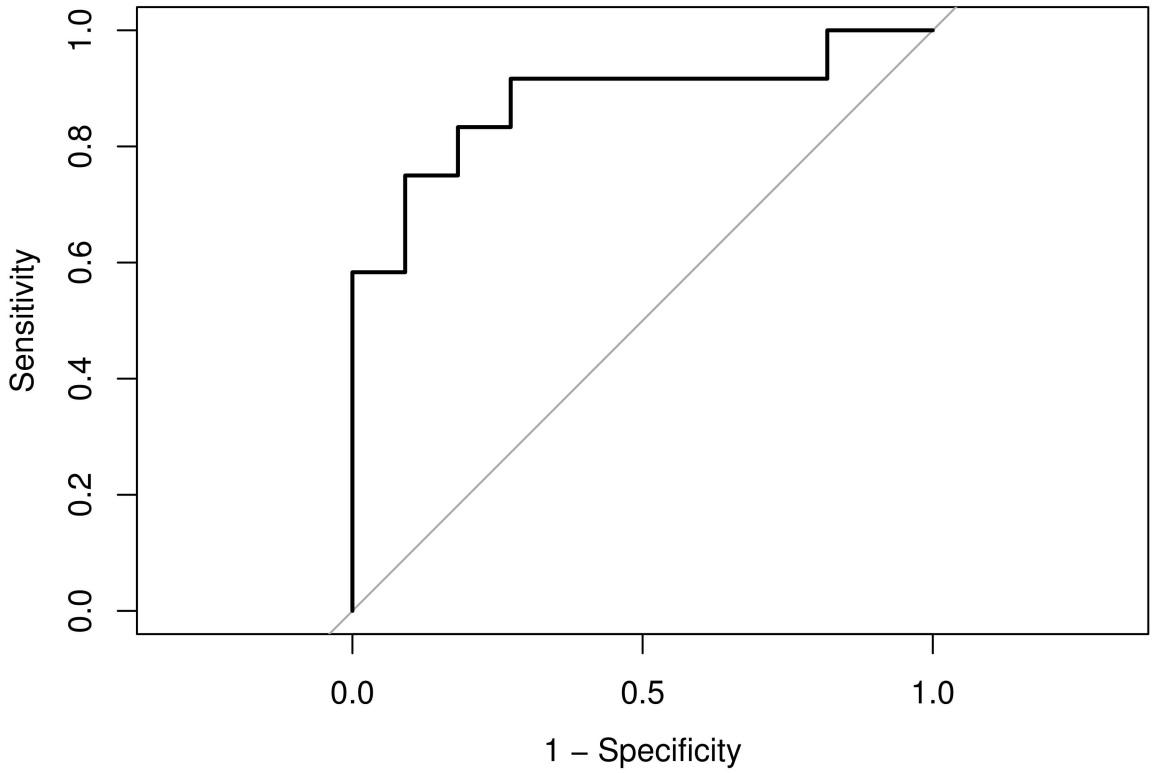
```r
# Specificity, TN/N = 0.8181818
con.mat[1]/sum(con.mat[1],con.mat[2])
```

```
## [1] 0.8181818
```

```r
# Overall misclassification, (FN+FP)/(N+P) = 0.173913
# or (1-sens)*[P/(P+N)]+(1-spec)*[N/(P+N)] = 0.173913
sum(con.mat[2],con.mat[3])/sum(con.mat)
```

```
## [1] 0.173913
```

```r
# ROC curve
# case = '+' (or nonbankrupt, 1) , control = '-' (or bankrupt, 0)
roc <- roc(test[, "Group"], prob, levels = c(0, 1))
plot(roc, legacy.axes = T)
```




```
lr_ful = c("LR full", sum(con.mat[2],con.mat[3])/sum(con.mat), roc$auc)

# Misclassification i higher than the previous model, as we would expect.
# It would appear that variable selection plays a
# large role in making a good model
# Our specificity has fallen due to this overfitted model
# but our sensitivity is the same.
# Also the ROC curve is further from the left corner
# which indicates a drop in performance using this model
```

3(c)

We perform same analysis as in 3(b),
but now using Linear Discriminant Analysis (LDA).

```
set.seed(1)
# question 3 part c ----
# create model with all predictors
library(MASS)
fit = lda(Group ~ ., data = train)

# Estimated probabilities for test data
# lda function already has the next steps within,
# so we can compute con.mat directly from this prob
```

```r
prob <- predict(fit, test)
# seems to result in same output with/without type = "response"

# Predicted classes (using 0.5 cutoff)
pred <- ifelse(prob$posterior[,2] >= 0.5, "nonbankrupt", "bankrupt")

# Test error rate
1 - mean(pred == test[, "Group"])
```

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

```r
# Confusion matrix and (sensitivity, specificity)
# `+' = nonbankrupt, `-' = bankrupt
con.mat = table(pred, test[, "Group"])
con.mat
```

```
##
## pred           0  1
##   bankrupt     8  2
##   nonbankrupt  3 10
```

```r
#       PRED CLASS
# TRUE     TN FP
# CLASS    FN TP
# Sensitivity, TP/P = 0.8333333
con.mat[4]/sum(con.mat[3],con.mat[4])
```

```
## [1] 0.8333333
```

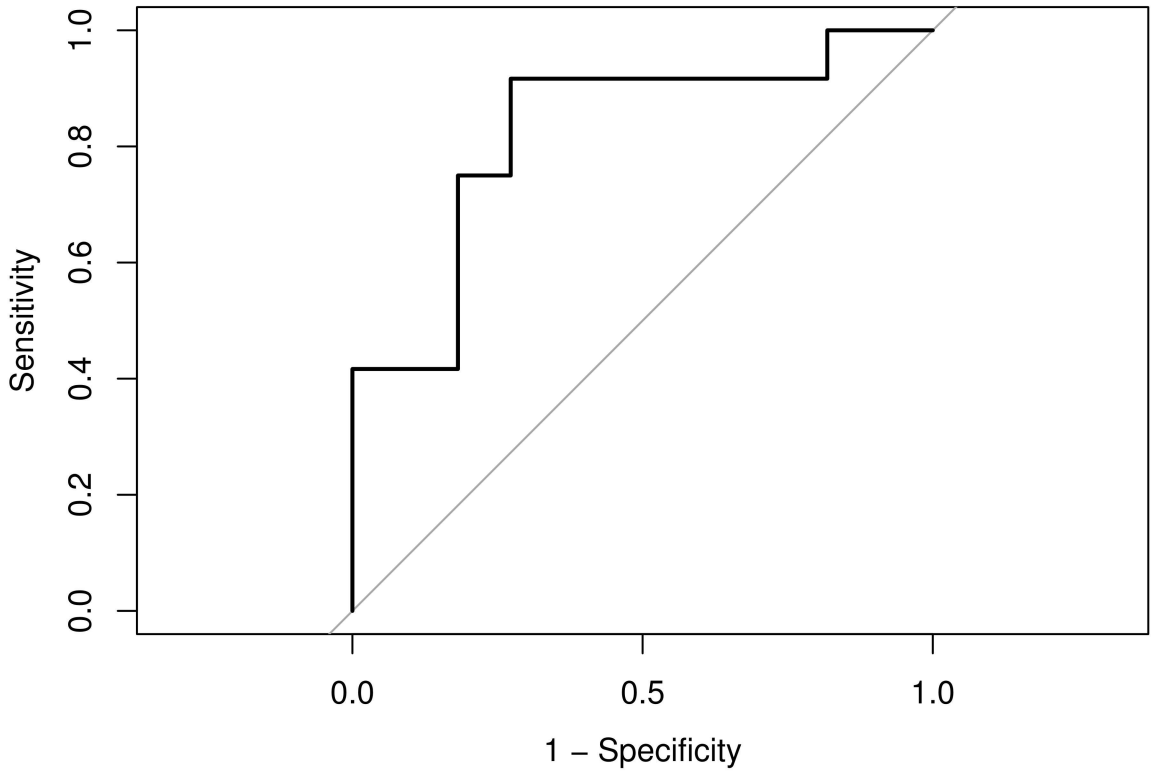
```r
# Specificity, TN/N = 0.7272727
con.mat[1]/sum(con.mat[1],con.mat[2])
```

```
## [1] 0.7272727
```

```r
# Overall misclassification, (FN+FP)/(N+P) = 0.2173913
# or (1-sens)*[P/(P+N)]+(1-spec)*[N/(P+N)] = 0.2173913
sum(con.mat[2],con.mat[3])/sum(con.mat)
```

```
## [1] 0.2173913
```

```r
# ROC curve
# case = '+' (or nonbankrupt, 1) , control = '-' (or bankrupt, 0)
library(pROC)
roc <- roc(test[, "Group"], prob$posterior[,2], levels = c(0, 1))
plot(roc, legacy.axes = T)
```




```
LDA = c("LDA", sum(con.mat[2],con.mat[3])/sum(con.mat), roc$auc)

# Sensitivity is the same once again, specificity has fallen even
# more from the previous model.  Misclassification is highest on LDA so far.
# This perhaps indicates a linear model is not a good fit for this data.
```

3(d)

Same analysis as previoius, but now using Quadratic Discriminant Analysis (QDA).

```
set.seed(1)
# question 3 part d ----
# create model with all predictors
library(MASS)
fit = qda(Group ~ ., data = train)

# Estimated probabilities for test data
# qda function already has the next steps within,
# so we can compute con.mat directly from this prob
prob <- predict(fit, test)
# seems to result in same output with/without type = "response"

# Predicted classes (using 0.5 cutoff)
pred <- ifelse(prob$posterior[,2] >= 0.5, "nonbankrupt", "bankrupt")
```

```r
# Test error rate
1 - mean(pred == test[, "Group"])
```

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

```r
# Confusion matrix and (sensitivity, specificity)
# `+' = nonbankrupt, `-' = bankrupt
con.mat = table(pred, test[, "Group"])
con.mat
```

```
##
## pred           0  1
##   bankrupt     9  2
##   nonbankrupt  2 10
```

```r
#       PRED CLASS
# TRUE     TN FP
# CLASS    FN TP
# Sensitivity, TP/P = 0.8333333
con.mat[4]/sum(con.mat[3],con.mat[4])
```

```
## [1] 0.8333333
```

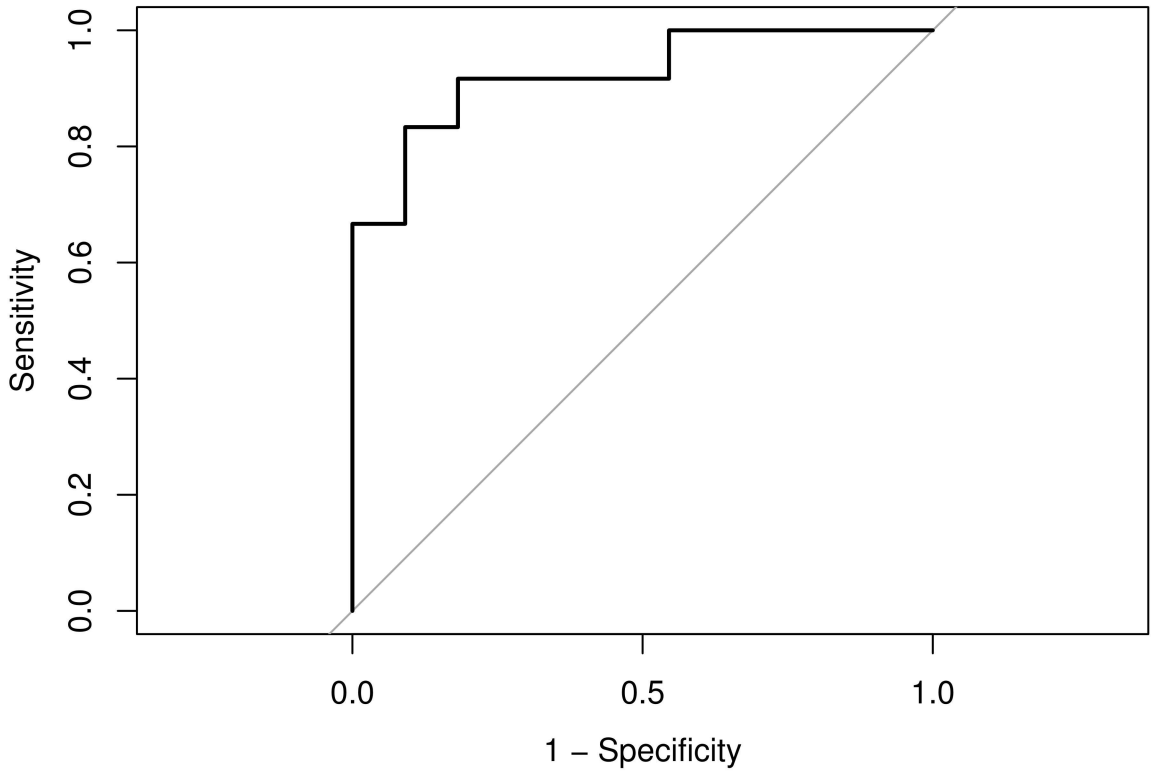
```r
# Specificity, TN/N = 0.8181818
con.mat[1]/sum(con.mat[1],con.mat[2])
```

```
## [1] 0.8181818
```

```r
# Overall misclassification, (FN+FP)/(N+P) = 0.173913
# or (1-sens)*[P/(P+N)]+(1-spec)*[N/(P+N)] = 0.173913
sum(con.mat[2],con.mat[3])/sum(con.mat)
```

```
## [1] 0.173913
```

```r
# ROC curve
# case = '+' (or nonbankrupt, 1) , control = '-' (or bankrupt, 0)
library(pROC)
roc <- roc(test[, "Group"], prob$posterior[,2], levels = c(0, 1))
plot(roc, legacy.axes = T)
```




```
QDA = c("QDA", sum(con.mat[2],con.mat[3])/sum(con.mat), roc$auc)

# Again Sensitivity is the same, interesting.  Specificity has come up
# from LDA model, looks to be where full logistic regression model is.
# Overall misclassification has gone back down as well,
# further exemplifying a linear model is not best for this data.
```

3(e)

Finally, we compare each model using Area under the curve and overall misclassification rate.

```
set.seed(1)
# question 3 part e ----
# From previous parts we look at who has the best AUC
# (area under the curve) and lowest misclassification
# To summarize here are the results
names = c("Model", "Misclassification Rate", "Area Under the Curve")
models = rbind(names, lr_red, lr_ful, LDA, QDA)
models
```

```
##        [,1]         [,2]                    [,3]
## names  "Model"      "Misclassification Rate" "Area Under the Curve"
## lr_red "LR reduced" "0.130434782608696"      "0.901515151515151"
```

```
## lr_ful "LR full"    "0.173913043478261"      "0.878787878787879"
## LDA    "LDA"        "0.217391304347826"      "0.825757575757576"
## QDA    "QDA"        "0.173913043478261"      "0.924242424242424"
```

```
# It appears that the reduced logistic regression model
# is the best selection of all choices
# due to having small misclassification rate and high AUC.
# However, QDA has an even higher AUC but at a cost of misclassification rate.
# This is a nice expectation because logistic regression is very good at
# modelling binary responses; we only have bankrupt and not bankrupt to fit.
# It also makes sense that the reduced model did better than the full model
# since it avoided overfitting problems typically associated with using
# all predictors.  This is nice because we can have inference on our
# coefficients and intuition on the predictors.
```