ROC Curves

Caravan data

```
# roc5.r
#
library(ROCR)      # prediction(), performance()
library(ISLR)      # Caravan Insurance Data
#
d0 = Caravan
dim(d0)
```

```
## [1] 5822   86
```

```
# 85 predictors, 1 response
#
# Response is Purchase
table(d0$Purchase)
```

```
##
##   No  Yes
## 5474  348
```

```
prop.table(table(d0$Purchase))
```

```
##
##         No        Yes
## 0.94022673 0.05977327
```

```
# only 6% people purchased insurance
#
y = d0$Purchase
X = d0[,-86]
#
# test set is 1st 1000 obs
#
test=1:1000
y.test=y[test]
x.test=X[test,]
#
# train set
#
y.train=y[-test]
x.train=X[-test,]
#
# logistic regression (no need to scale data)
#
model1= glm(Purchase~.,Caravan,family=binomial,subset=-test)
# get probabs for test set
probabs1 = predict(model1,Caravan[test,],type="response")
head(probabs1)
```

```
##          1          2          3          4          5          6
```

```
## 0.092147333 0.009350753 0.057483558 0.091716480 0.026093327 0.017693380
```

```
#
contrasts(y)
```

```
##     Yes
## No    0
## Yes   1
```

```
# "Positive" outcome is Yes
#
# Predicted category is "Yes" if posterior probab > 0.5
#
yhat = rep("No",1000)
yhat[probabs1 > 0.5] = "Yes"
table("test"=y.test,"prediction"=yhat)
```

```
##      prediction
## test   No Yes
##   No  934   7
##   Yes  59   0
```

```
#
# Confusion Matrix
confusionmat = as.matrix(table(y.test,yhat))
rowSums(confusionmat)
```

```
##  No Yes
## 941  59
```

```
TPR1 = confusionmat[2,2]/rowSums(confusionmat)[2]
TPR1
```

```
## Yes
##   0
```

```
FPR1 = confusionmat[1,2]/rowSums(confusionmat)[1]
FPR1
```

```
##          No
## 0.007438895
```

```
#
# Predicted category is "Yes" if posterior probab > 0.25
#
yhat = rep("No",1000)
yhat[probabs1 > 0.25] = "Yes"
table("test"=y.test,"prediction"=yhat)
```

```
##      prediction
## test   No Yes
##   No  919  22
##   Yes  48  11
```

```
#
# Confusion Matrix
confusionmat = as.matrix(table(y.test,yhat))
rowSums(confusionmat)
```

```
##  No Yes
## 941  59
```

```r
TPR2 = confusionmat[2,2]/rowSums(confusionmat)[2]
TPR2
```

```
##       Yes
## 0.1864407
```

```r
FPR2 = confusionmat[1,2]/rowSums(confusionmat)[1]
FPR2
```

```
##         No
## 0.02337938
```

```r
#
# loop for ROC Curve
#
cutoff = seq(0.001,0.92,0.001)
n = length(cutoff)
n
```

```
## [1] 920
```

```r
TPR = rep(0,n)
FPR = rep(0,n)
#
for(i in cutoff)
{
  yhat = rep("No",1000)
  yhat[probabs1 > i] = "Yes"
  confusionmat = as.matrix(table(y.test,yhat))
  j = n*i
  TPR[j] = confusionmat[2,2]/rowSums(confusionmat)[2]
  FPR[j] = confusionmat[1,2]/rowSums(confusionmat)[1]
}
#
df1 = data.frame(cutoff,TPR,FPR)
head(df1,15)
```

```
##    cutoff       TPR       FPR
## 1   0.001 0.9830508 0.9819341
## 2   0.002 0.9830508 0.9734325
## 3   0.003 0.9830508 0.9638682
## 4   0.004 0.9830508 0.9489904
## 5   0.005 0.9830508 0.9362380
## 6   0.006 0.9830508 0.9234857
## 7   0.007 0.9830508 0.9011690
## 8   0.008 0.9830508 0.8799150
## 9   0.009 0.9830508 0.8480340
## 10  0.010 0.9661017 0.8214665
## 11  0.011 0.9322034 0.7800213
## 12  0.012 0.9322034 0.7608927
## 13  0.013 0.9322034 0.7438895
## 14  0.014 0.9152542 0.7290117
## 15  0.015 0.9152542 0.7173220
```

```r
which(df1$cutoff == 0.50,)
```

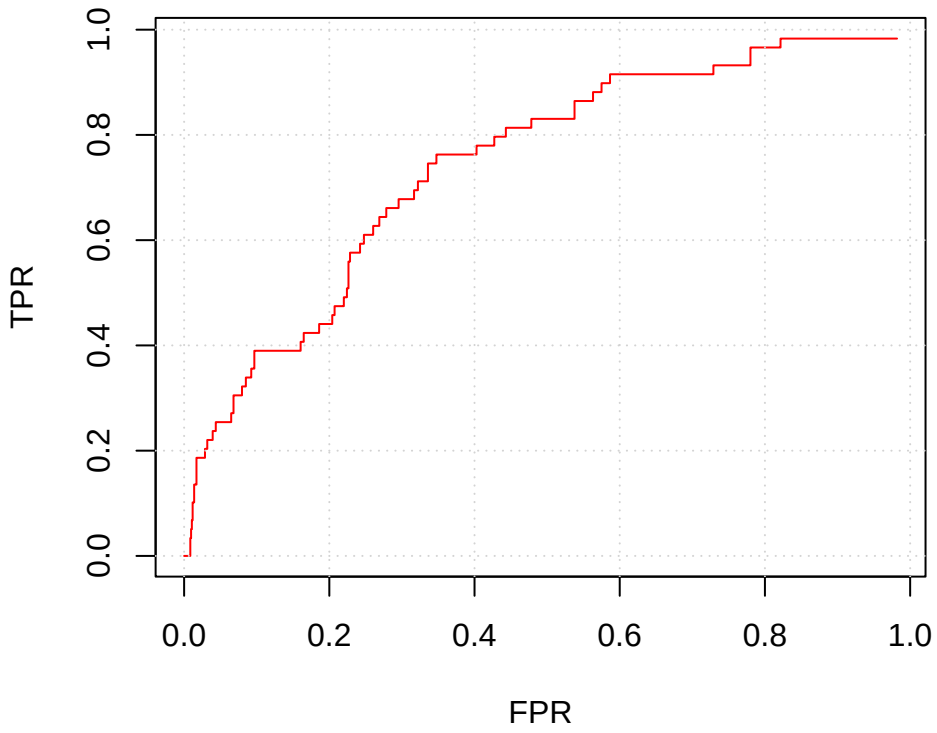
```
## [1] 500
```

```r
df1[500,]
```

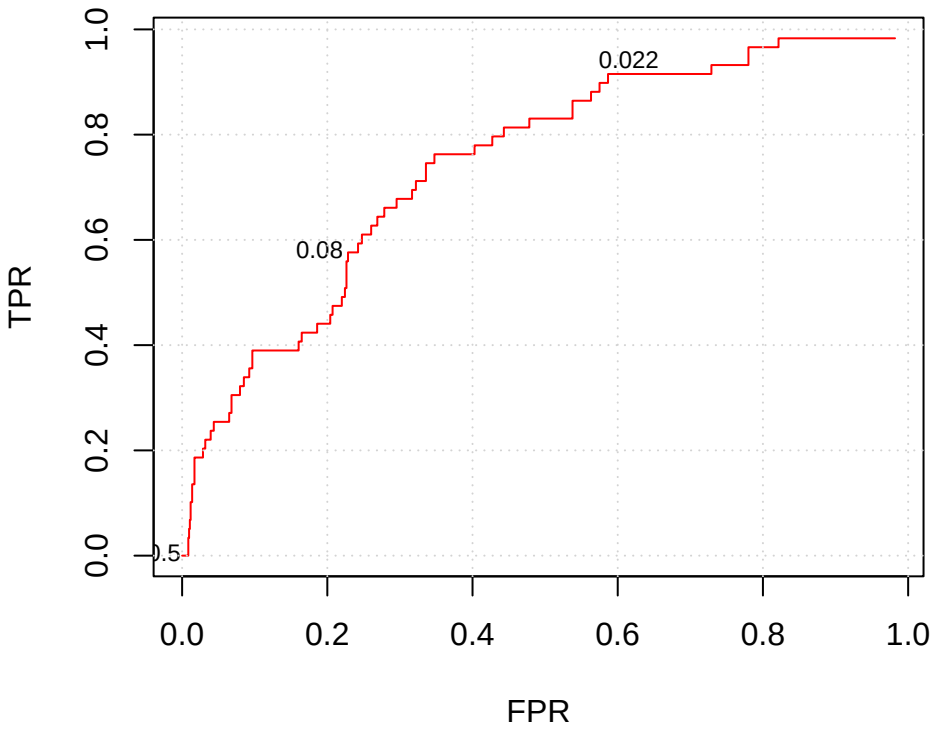
```
##     cutoff TPR         FPR
```

```
## 500    0.5    0 0.006376196
```

```r
plot(FPR,TPR,type="s",col="red")
grid()
```



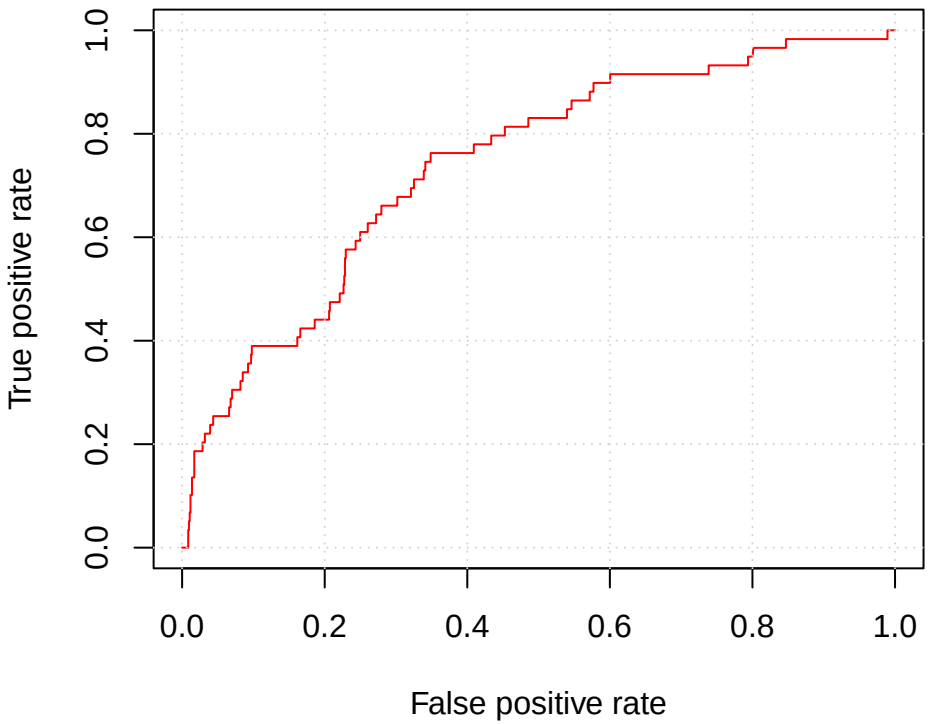

```r
#
# text(FPR,TPR,labels = rownames(df1),pos=2,cex=0.5,offset=0.15)
# best cutoff values
plot(FPR,TPR,type="s",col="red")
text(FPR[22],TPR[22],labels = df1$cutoff[22],pos=3,cex=0.75,offset=0.15)
text(FPR[80],TPR[80],labels = df1$cutoff[80],pos=2,cex=0.75,offset=0.15)
text(FPR[500],TPR[500],labels = df1$cutoff[500],pos=2,cex=0.75,offset=0.15)
grid()
```



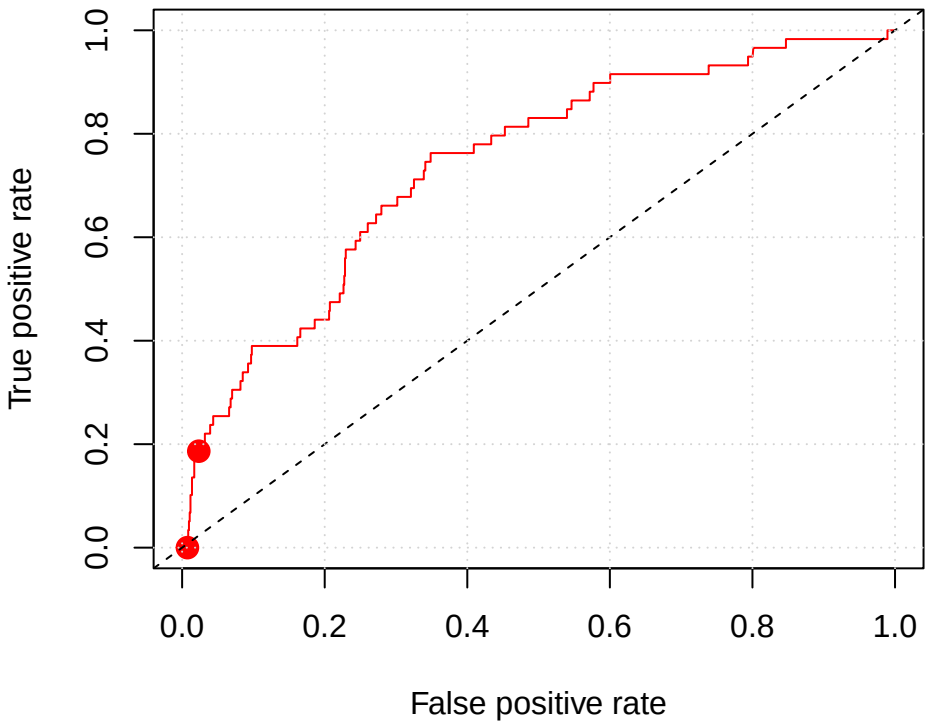

```
#
df1[c(50,80,500),]
```

```
##     cutoff       TPR         FPR
## 50    0.05 0.7457627 0.347502657
## 80    0.08 0.5762712 0.229543039
## 500   0.50 0.0000000 0.006376196
```

```
#
# use library ROCR to plot ROC curve
pred_ROCR = prediction(probabs1,y.test)
roc_ROCR  = performance(pred_ROCR,
                        measure="tpr",
                        x.measure="fpr")
#
# plot ROC (x-axis: fpr, y-axis = tpr)
#
plot(roc_ROCR,col="red")
grid()
```



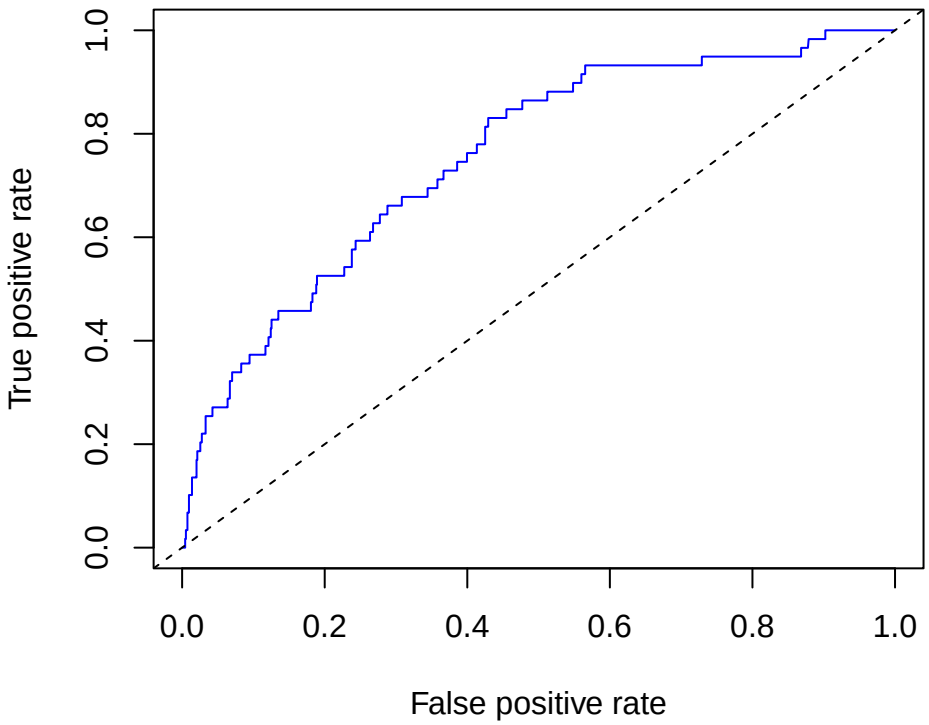

```
#
plot(roc_ROCR,col="red")
points(FPR1,TPR1,col="red",cex=1.5,pch=19)
points(FPR2,TPR2,col="red",cex=1.5,pch=19)
abline(a = 0, b = 1,lty=2)
grid()
```



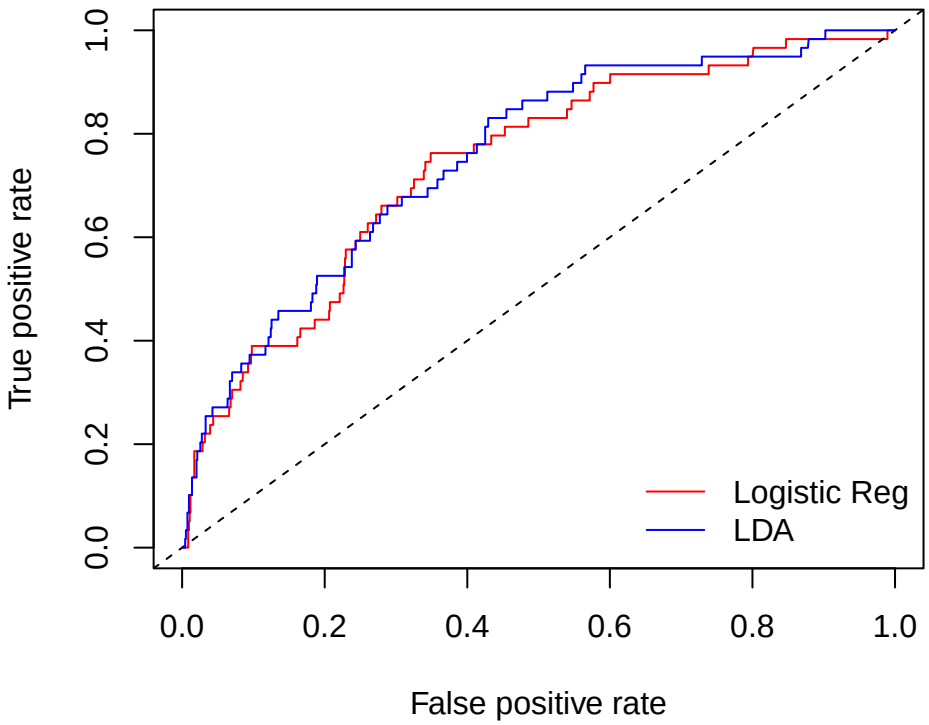

```
#
# Discriminant Analysis
#
library(MASS)
model2  = lda(Purchase~.,data = Caravan,
              subset=-test)
probabs = predict(model2,Caravan[test,])
head(probabs$posterior,4)
```

```
##          No        Yes
## 1 0.9379531 0.06204693
## 2 0.9896201 0.01037994
## 3 0.9552795 0.04472047
## 4 0.9318285 0.06817154
```

```
probabs2 <- probabs$posterior[,2]
pred_ROCR2 = prediction(probabs2,y.test)
roc_ROCR2 = performance(pred_ROCR2,
                        measure="tpr",
                        x.measure="fpr")
#
# Add ROC for LDA
#
plot(roc_ROCR2,col="blue")
abline(a = 0, b = 1,lty=2)
```




```
#
plot(roc_ROCR,col="red")
abline(a = 0, b = 1,lty=2)
plot(roc_ROCR2,col="blue",add = T)
legend("bottomright",
       legend = c("Logistic Reg", "LDA"),
       col = c("red", "blue"),
       lty = c(1,1), bty = "n")
```




```
#
# AUC - logistic regression
auc1 <- performance(pred_ROCR, measure = "auc")
auc1 <- auc1@y.values[[1]]
auc1
```

```
## [1] 0.7407464
```

```
#
# AUC - Linear Discriminant Analysis
auc2 <- performance(pred_ROCR2, measure = "auc")
auc2 <- auc2@y.values[[1]]
auc2
```

```
## [1] 0.7542733
```