Homework 4

Yue Peng

a. Construct a good logistic regression model predicting the decision to switch wells as a function of the 4 predictors (arsenic, distance, association and education) on the training data. Consider potential transformations of continuous variables and possible interactions.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.19158    0.07779  -2.463   0.0138 *
arsenic      0.35803    0.04220   8.485   <2e-16 ***
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.46773    0.05438   8.601   <2e-16 ***
assoc1      -0.17631    0.08184  -2.154   0.0312 *
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.21619    0.06133   3.525 0.000423 ***
educ         0.03843    0.01025   3.748 0.000178 ***
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.712150   0.066554  10.700  < 2e-16 ***
dist        -0.006467   0.001049  -6.162 7.18e-10 ***
```

After fitting each predictor into the logistic regression model, we decided to use them all.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.002081   0.108741   0.019   0.9847
arsenic      0.454297   0.045877   9.903  < 2e-16 ***
assoc1      -0.170160   0.084566  -2.012   0.0442 *
educ         0.042100   0.010589   3.976 7.01e-05 ***
dist        -0.009329   0.001132  -8.238  < 2e-16 ***
```

We would try to remove the “intercept” term since the p-value was way larger than 0.05.

```
Coefficients:
         Estimate Std. Error z value Pr(>|z|)
arsenic  0.454297   0.045877   9.903  < 2e-16 ***
educ     0.042100   0.010589   3.976 7.01e-05 ***
dist    -0.009329   0.001132  -8.238  < 2e-16 ***
assoc0   0.002081   0.108741   0.019    0.985
assoc1  -0.168079   0.110410  -1.522    0.128
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We would try to remove the “assoc” term since the p-value was larger than 0.05.

```
Coefficients:
         Estimate Std. Error z value Pr(>|z|)
arsenic  0.435559   0.037484  11.620  < 2e-16 ***
educ     0.038272   0.009178   4.170 3.04e-05 ***
dist    -0.009590   0.001059  -9.052  < 2e-16 ***
```

```
Analysis of Deviance Table

Model 1: switch ~ -1 + arsenic + educ + dist + assoc
Model 2: switch ~ -1 + arsenic + educ + dist
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      2515     3227.0
2      2517     3231.6 -2  -4.6554  0.09752 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

With the ANOVA table, we decided to use “arsenic”, “educ” and “dist” as predictors since the p-value was larger than 0.05.

The next step was to test the interaction term.

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
arsenic       0.370024   0.042779   8.650  < 2e-16 ***
educ          0.004483   0.014532   0.308  0.75771
dist         -0.008946   0.001083  -8.262  < 2e-16 ***
arsenic:educ  0.028594   0.009673   2.956  0.00312 **

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
arsenic    0.4870470  0.0414008  11.764  < 2e-16 ***
educ       0.0098986  0.0125541   0.788  0.43042
dist      -0.0126871  0.0014512  -8.742  < 2e-16 ***
educ:dist  0.0007718  0.0002380   3.243  0.00118 **

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
arsenic       0.4409230  0.0454797   9.695  < 2e-16 ***
dist         -0.0093432  0.0015865  -5.889 3.88e-09 ***
educ          0.0374955  0.0099013   3.787 0.000153 ***
arsenic:dist -0.0001670  0.0007994  -0.209 0.834494
```

We decided to test the interaction effect of “educ” with “arsenic” and with “dist”.

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
arsenic       0.4317063  0.0536099   8.053 8.10e-16 ***
educ         -0.0024227  0.0148713  -0.163   0.8706
dist         -0.0113854  0.0016510  -6.896 5.34e-12 ***
educ:dist     0.0005533  0.0002751   2.012   0.0443 *
arsenic:educ  0.0172431  0.0111076   1.552   0.1206

Analysis of Deviance Table

Model 1: switch ~ -1 + arsenic + educ * dist
Model 2: switch ~ -1 + arsenic + educ * dist + educ * arsenic
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      2516     3220.9
2      2515     3218.4  1   2.4596   0.1168
```

The model with only “educ*dist” was better since the p-value was larger than 0.05.

```
Analysis of Deviance Table

Model 1: switch ~ -1 + arsenic + educ + dist
Model 2: switch ~ -1 + arsenic + educ * dist
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      2517     3231.6
2      2516     3220.9  1   10.767 0.001033 **
```

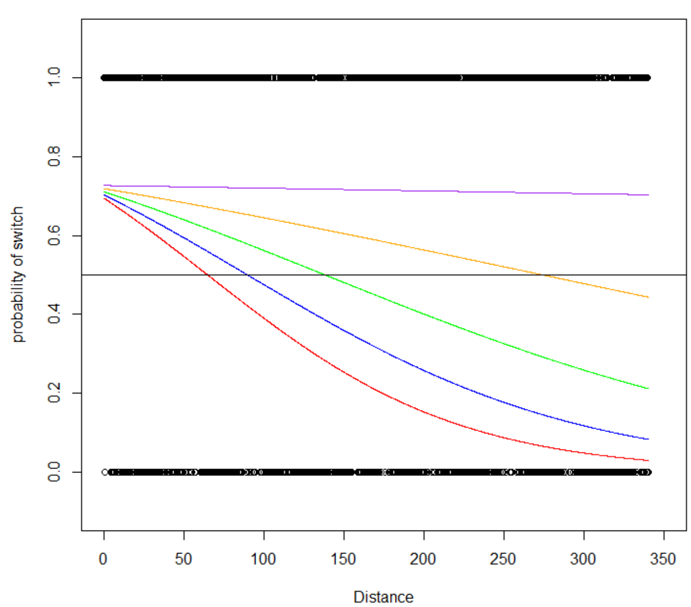
The model including the "educ*dist" was better than model without interaction term since the p-value was less than 0.05.

Thus, our final model was

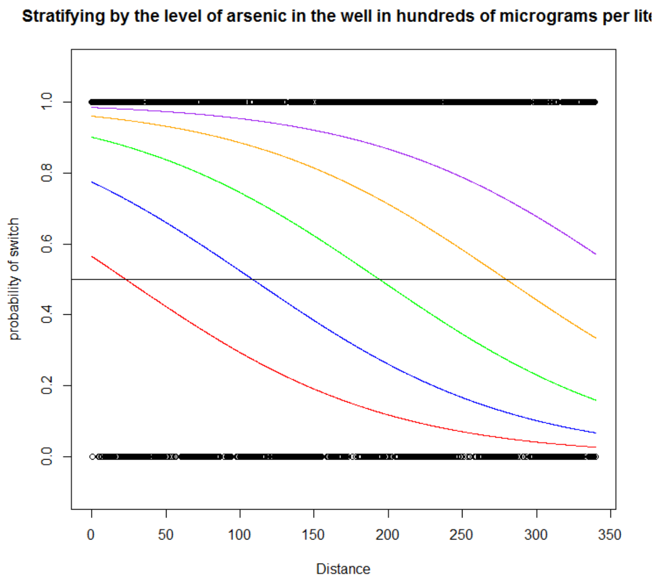
$$\log\left(\frac{p}{1-p}\right) = 0.4870 \times arsenic + 0.0099 \times educ - 0.0127 \times dist + 0.0008 \times educ \times dist$$

b. Compute and graph the predicted probabilities stratifying by the predictors. You could do this using graph such as in the papers we discussed in class or by using contour plots which would allow you to graph two continuous predictors on the same plot. You can array different lines and plots to try to put this all on one sheet or you can spread across different plots. See what works best.

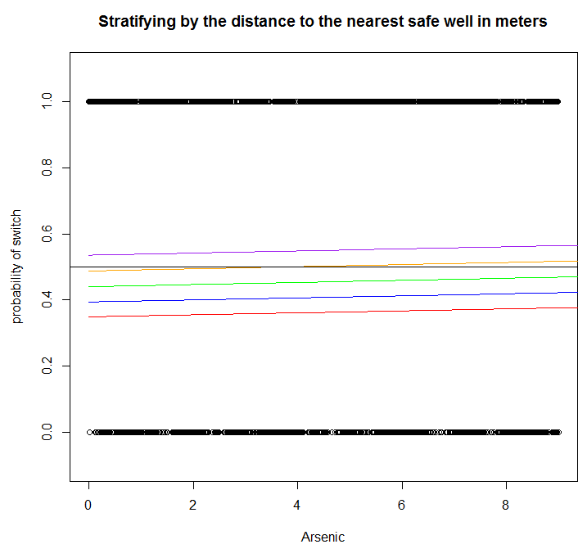
Stratifying by the number of years of education of the head of household.



For the curves above, the "educ" kept increases while fixing the "arsenic" as the mean value of the training dataset. The curve with longer "educ" went above the one with shorter "educ". It meant that the larger number of years of education of the head of household, the more likely for the household to switch wells.



For the curves above, the “arsenic” kept increases while fixing the “educ” as the mean value of the training dataset. The curve with higher “arsenic” went above the one with lower “arsenic”. It meant that the higher level of arsenic in the well in hundreds of micrograms per liter, the more likely for the household to switch wells.



For the curves above, the “educ” kept increases while fixing the “dist” as the mean value of the training dataset. The curve with longer “educ” went above the one with shorter “educ”. It meant that the larger number of years of education of the head of household, the more likely for the household to switch wells.

c. Compute the confusion matrix on the test data using p = 0.5 as a cutoff and discuss what this tells you about the predictive model you have constructed (e.g. sensitivity, specificity, error rate, etc.)

	True switch	True no switch
Predicted switch	210 (TP)	208 (FP)
Predicted no switch	24 (FN)	58 (TN)

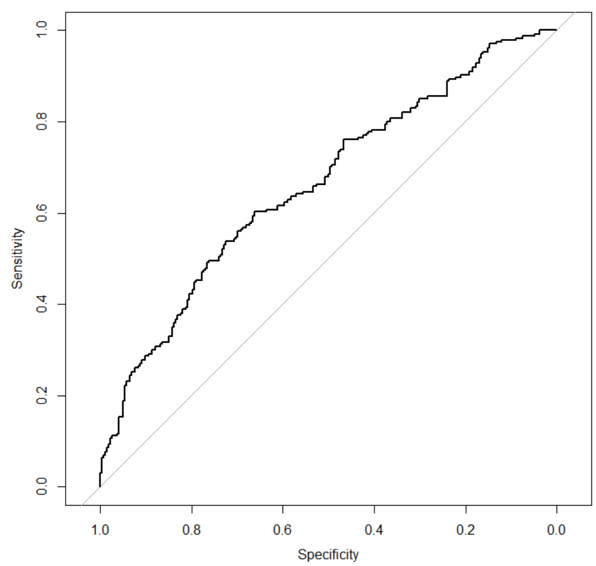
$$\text{Sensitivity} = \frac{210}{210 + 24} = 0.897$$

$$\text{Specificityty} = \frac{58}{208 + 58} = 0.218$$

$$\text{Error Rate} = 1 - \frac{210 + 58}{210 + 24 + 208 + 58} = 0.464$$

According to the high sensitivity and low specificity, this model performed well in predicting the household has switched wells but badly in predicting the household has not switched wells.

d. Construct an ROC plot and compute the area under the ROC curve.



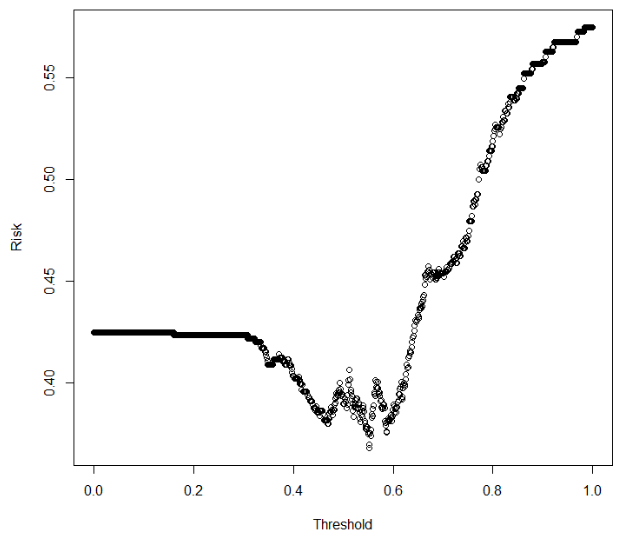

```
Call:
roc.default(response = test_dat$switch, predictor = prob, plot = T)

Data: prob in 266 controls (test_dat$switch 0) < 234 cases (test_dat$switch 1).
Area under the curve: 0.6622
```

The ROC curve was shown above, and the AUC was 0.6622, which did relatively well in prediction.

e. What does this curve tell you about choice of threshold that balances sensitivity with specificity (i.e., how would you balance risk of switching and not switching?)

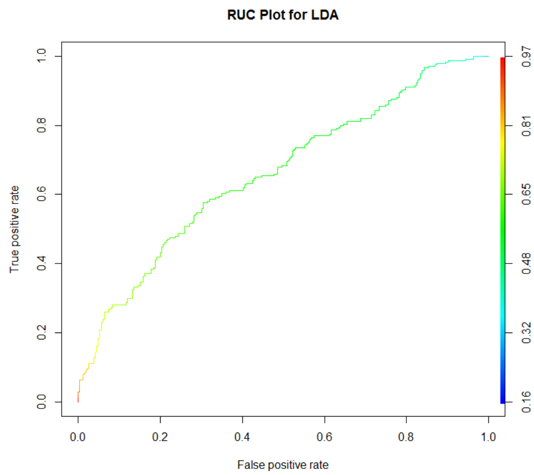
$$\text{risk} = prob_{True\ switch} \times (1 - sensitivity) + (1 - prob_{True\ switch}) \times (1 - specificity)$$



The threshold with the least risk (0.368) was 0.55. If refitting this into our previous model, our sensitivity would be changed into 0.76 and specificity was 0.45. Compared to the previous model, the predictive ability increased since the specificity has increased a lot while sacrifice a little sensitivity.

f. Repeat this analysis using linear discriminant analysis, quadratic discriminant analysis and K nearest neighbor with K = 1 and K = 5. For discriminant analysis, note that the predict function returns 3 elements: *class* is a binary indicator as to whether the posterior probability is greater than 0.5, *posterior* gives the posterior predictive probability and *x* contains the linear discriminant for the LDA function (missing for QDA). Note that for KNN, you will only get classifications, not probabilities.

1. Linear Discriminant Analysis



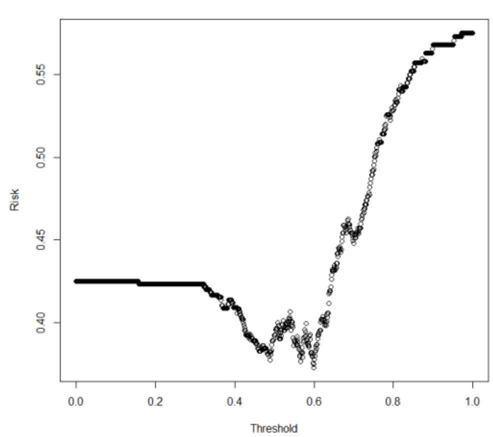
	True switch	True no switch
Predicted switch	218 (TP)	221 (FP)
Predicted no switch	16 (FN)	45 (TN)

$$\text{Sensitivity} = \frac{218}{218+16} = 0.932$$

$$\text{Specificityty} = \frac{45}{221+45} = 0.169$$

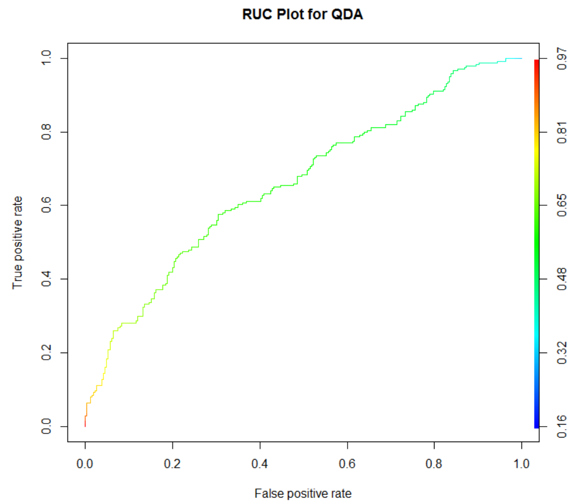
$$\text{Error Rate} = 1 - \frac{218+45}{218+16+221+45} = 0.474$$

AUC is 0.6607



The threshold with the least risk (0.373) was 0.5996. If refitting this into our previous model, our sensitivity would be changed into 0.58 and specificity was 0.70. Compared to the previous model, the predictive ability cannot tell whether exactly increasing or not since the specificity has increased a lot but also sacrifice a lot sensitivity.

2. Quadratic Discriminant Analysis



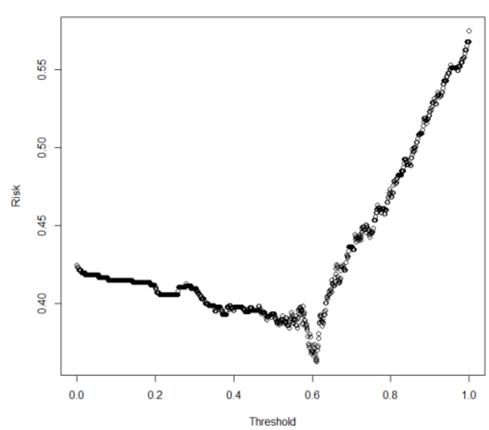
	True switch	True no switch
Predicted switch	220 (TP)	225 (FP)
Predicted no switch	14 (FN)	41(TN)

$$\text{Sensitivity} = \frac{220}{220 + 14} = 0.940$$

$$\text{Specificityty} = \frac{41}{225 + 41} = 0.154$$

$$\text{Error Rate} = 1 - \frac{220 + 41}{220 + 14 + 225 + 41} = 0.478$$

AUC is 0.6601



The threshold with the least risk (0.363) was 0.61. If refitting this into our previous model, our sensitivity would be changed into 0.65 and specificity was 0.62. Compared to the previous model, the predictive ability increased since the specificity has increased a lot and almost the same as sensitivity.

3. KNN

When K=1,

	True switch	True no switch
Predicted switch	208 (TP)	51(FP)
Predicted no switch	26 (FN)	215(TN)

$$\text{Sensitivity} = \frac{208}{208 + 26} = 0.889$$

$$\text{Specificityty} = \frac{215}{215 + 51} = 0.808$$

$$\text{Error Rate} = 1 - \frac{208 + 215}{208 + 215 + 51 + 26} = 0.154$$

AUC is 0.85

According to the high sensitivity and high specificity, this model performed well in both predicting the household has switched wells and in predicting the household has not switched wells.

When K=5,

	True switch	True no switch
Predicted switch	212 (TP)	78(FP)
Predicted no switch	22 (FN)	188(TN)

$$\text{Sensitivity} = \frac{212}{212 + 22} = 0.906$$

$$\text{Specificityty} = \frac{188}{188 + 78} = 0.707$$

$$\text{Error Rate} = 1 - \frac{212 + 188}{212 + 188 + 78 + 22} = 0.20$$

AUC is 0.81

According to the high sensitivity and high specificity, this model performed well in both predicting the household has switched wells and in predicting the household has not switched wells. But it was a bit worse than K=1 case.

Appendix

```
# Homework 4
# Read the dataset
setwd("D:/Dropbox (Brown)/Fall2017/PHP2550/HW4")
dat <- read.table("wells.txt", sep = " ", header = T)
# switch is a binary indicator for whether the household switched wells
dat$switch <- as.factor(dat$switch)
# arsenic is the level of arsenic in the well in hundreds of micrograms per liter

# dist is the distance to the nearest safe well in meters

# assoc is whether household members are active in community organizations
dat$assoc <- as.factor(dat$assoc)
# educ is the number of years of education of the head of household.
str(dat)
# Separate into train and test set
train_dat <- dat[1:2520,]
test_dat <- dat[2521:3020, ]

### a.
# Test each predictor
summary(glm(switch ~ arsenic, family = binomial(link = logit), data = train_dat))
summary(glm(switch ~ assoc, family = binomial(link = logit), data = train_dat))
summary(glm(switch ~ educ, family = binomial(link = logit), data = train_dat))
summary(glm(switch ~ dist, family = binomial(link = logit), data = train_dat))

m0 <- glm(switch ~ arsenic + assoc + educ + dist, family = binomial(link = logit), data = train_dat)
summary(m0)

m1 <- glm(switch ~ -1 + arsenic + educ + dist + assoc, family = binomial(link = logit), data = train_dat)
summary(m1)

m2 <- glm(switch ~ -1 + arsenic + educ + dist, family = binomial(link = logit), data = train_dat)
summary(m2)
anova(m1, m2,test = "Chisq")

m3 <- glm(switch ~ -1 + arsenic * educ + dist, family = binomial(link = logit), data = train_dat)
summary(m3)

m4 <- glm(switch ~ -1 + arsenic +    educ * dist, family = binomial(link = logit), data = train_dat)
summary(m4)
```

```
m5 <- glm(switch ~ -1 + arsenic * dist + educ, family = binomial(link = logit), data = train_dat)
summary(m5) # not significant

m6 <- glm(switch ~ -1 + arsenic +    educ * dist +educ*arsenic, family = binomial(link = logit), data =
train_dat)
summary(m6)
anova(m2, m6,test = "Chisq")

model <- glm(switch ~ -1 + arsenic +    educ * dist, family = binomial(link = logit), data = train_dat)
summary(model)
anova(m2,model,test = "Chisq")
# b
p <- function(z){
  return(exp(z)/(1+exp(z)))
}

z <- function(arsenic, educ, dist){
  return(coef(model)[1]*arsenic+coef(model)[2]*educ+coef(model)[3]*dist+coef(model)[4]*educ*dist)
}
# EDUC
range(train_dat$dist)
x <- seq(0, 340, length.out = 2520)
plot(x, as.numeric(as.character(train_dat$switch)),xlim=c(0,350), ylim = c(-0.1,1.1), xlab = "Distance",
     ylab="probability of switch", main = c("Stratifying by the number of years of education of the head of
household."))
range(train_dat$educ)
educs <- c(0,4,8,12,16)
cols <- c("red", "blue", "green", "orange", "purple")
for (i in seq_along(educs)){
lines(x, p(z(mean(train_dat$arsenic),educ = educs[i], x)), col=cols[i])
}
abline(h=0.5)
# ARSENIC
plot(x, as.numeric(as.character(train_dat$switch)), ylim = c(-0.1,1.1), xlab = "Distance", ylab="probability of
switch",
     main = c("Stratifying by the level of arsenic in the well in hundreds of micrograms per liter"))
range(train_dat$arsenic)
arsenics <- c(0.5, 2.5, 4.5, 6.5, 8.5)
for (i in seq_along(educs)){
  lines(x, p(z(arsenics[i],educ = mean(train_dat$arsenic), x)), col=cols[i])
```

```
}
abline(h=0.5)
# DIST
x1 <- seq(0, 9, length.out = 2520)
plot(x1, as.numeric(as.character(train_dat$switch)), ylim = c(-0.1,1.1), xlab = "Arsenic", ylab="probability of switch",
        main = c("Stratifying by the distance to the nearest safe well in meters"))

for (i in seq_along(educs)){
  lines(x, p(z(x1,educ = educs[i], mean(train_dat$dist))), col=cols[i])
}
abline(h=0.5)
#c
prob <- predict.glm(model, type = "response", newdata = test_dat)
pred <- ifelse(prob>0.5,1,0)
table(pred, test_dat$switch)
# Sensitivity
210/(24+210)
sensitivity(confusion.matrix(test_dat$switch, prob, threshold = 0.5))
# Specificity
58/(208+58)
specificity(confusion.matrix(test_dat$switch, prob, threshold = 0.5))
# 1- Accuracy
1-(210+58)/(58+24+208+210)

# d
library(pROC)
roc(test_dat$switch, prob,plot = T)

# e
library(SDMTools)
p_ts <- sum(dat$switch==1)/dim(dat)[1]
risk <- function(p){
  cm <- confusion.matrix(test_dat$switch, prob, threshold = p)
  sen <- sensitivity(cm)
  spe <- specificity(cm)
  return(p_ts*(1-sen)+(1-p_ts)*(1-spe))
}

x2 <- seq(0,1, length.out = 1000)
y <- mapply(risk, x2)
```

```
plot(x2, y, xlab = "Threshold", ylab = "Risk")

# Threshold with least risk
min(y)
threshold <-x2[which.min(y)]
threshold

cm0 <- confusion.matrix(test_dat$switch, pred, threshold = threshold)
sensitivity(cm0)
specificity(cm0)

# f
# LDA
library(MASS)
library(ROCR)
mlda0 <- lda(switch ~ -1 + arsenic +    educ * dist, train_dat)
mlda0_fit <- predict(mlda0, test_dat)
prob_lda <- predict(mlda0, test_dat)$posterior[,2]
pred_lda <- prediction(prob_lda, test_dat$switch)
perf_lda <- performance(pred_lda, "tpr", "fpr")
plot(perf_lda,main="RUC Plot for LDA", colorize=TRUE)
table(mlda0_fit$class, test_dat$switch)
performance(pred_lda, "auc")@y.values

p_ts <- sum(dat$switch==1)/dim(dat)[1]
risk <- function(p){
   cm <- confusion.matrix(test_dat$switch, prob_lda, threshold = p)
   sen <- sensitivity(cm)
   spe <- specificity(cm)
   return(p_ts*(1-sen)+(1-p_ts)*(1-spe))
}

x3 <- seq(0,1, length.out = 1000)
y <- mapply(risk, x3)
plot(x3, y, xlab = "Threshold", ylab = "Risk")

# Threshold with least risk
min(y)
threshold <-x3[which.min(y)]
threshold
```

```
cm1 <- confusion.matrix(test_dat$switch, prob_lda, threshold = threshold)
sensitivity(cm1)
specificity(cm1)

#QDA
mqda0 <- qda(switch ~ -1 + arsenic +    educ * dist, train_dat)
mqda0_fit <- predict(mqda0, test_dat)
prob_qda <- predict(mqda0, test_dat)$posterior[,2]
pred_qda <- prediction(prob_qda, test_dat$switch)
perf_qda <- performance(pred_qda, "tpr", "fpr")
plot(perf_lda,main="RUC Plot for QDA", colorize=TRUE)
table(mqda0_fit$class, test_dat$switch)
performance(pred_qda, "auc")@y.values

p_ts <- sum(dat$switch==1)/dim(dat)[1]
risk <- function(p){
   cm <- confusion.matrix(test_dat$switch, prob_qda, threshold = p)
   sen <- sensitivity(cm)
   spe <- specificity(cm)
   return(p_ts*(1-sen)+(1-p_ts)*(1-spe))
}

x4 <- seq(0,1, length.out = 1000)
y <- mapply(risk, x4)
plot(x4, y, xlab = "Threshold", ylab = "Risk")

# Threshold with least risk
min(y)
threshold <-x4[which.min(y)]
threshold

cm2 <- confusion.matrix(test_dat$switch, prob_qda, threshold = threshold)
sensitivity(cm2)
specificity(cm2)


# KNN
mknn0 <- knn(train = train_dat, test = test_dat,cl=train_dat$switch, k=1)
table(mknn0, test_dat$switch)
multiclass.roc(response=test_dat$switch, predictor = as.ordered(mknn0))
```

```
mknn1 <- knn(train = train_dat, test = test_dat,cl=train_dat$switch, k=5)
table(mknn1, test_dat$switch)
multiclass.roc(response=test_dat$switch, predictor = as.ordered(mknn1))
```