HW

Amin Yakubu

4/7/2019

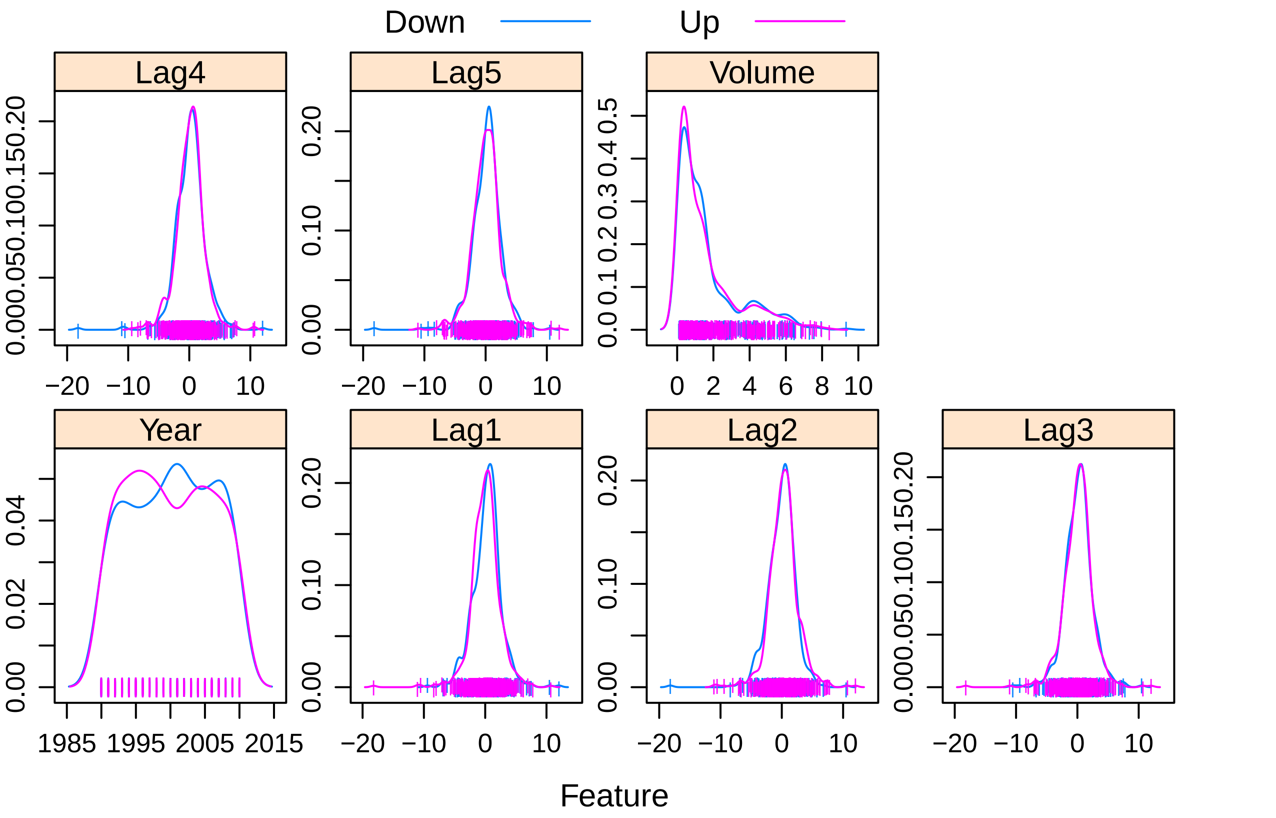
```
library(ISLR)
library(caret)
library(corrplot)
library(pROC)
library(MASS)
```

Question a

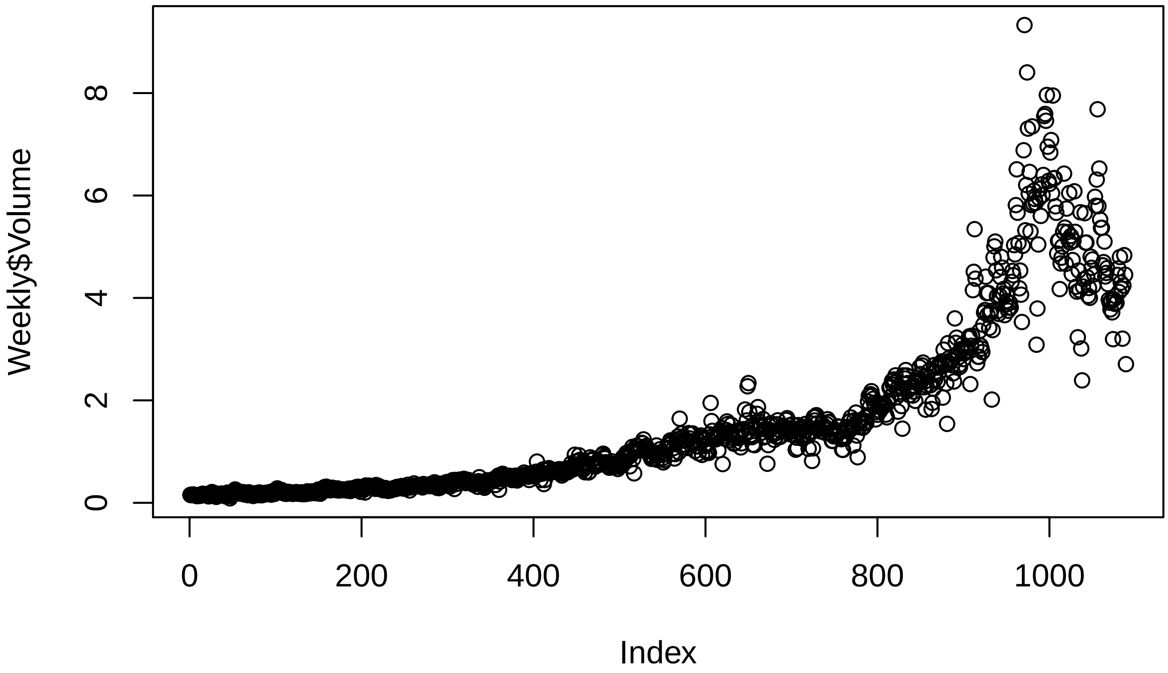
Produce some graphical summaries of the Weekly data

```
data(Weekly)

# excluding Today as a predictor
Weekly = Weekly[,-8]
```

```
featurePlot(x = Weekly[, 1:7],
            y = Weekly$Direction,
            scales = list(x = list(relation = "free"), #  because year is on a different scale
                        y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```



```r
plot(Weekly$Volume)
```



Here we see that volume is increasing over time. Also, we see that year is highly correlated with volume. From the graphs we can see that lag1-lag5 are approximately normally distributed. Volume is skewed to the right.

```r
cor(Weekly[,-8])
```

```
##               Year         Lag1        Lag2        Lag3        Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.03112792
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.07127388
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.05838153
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.07539587
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.00000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.07567503
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.06107462
##                Lag5      Volume
## Year   -0.030519101  0.84194162
## Lag1   -0.008183096 -0.06495131
## Lag2   -0.072499482 -0.08551314
## Lag3    0.060657175 -0.06928771
## Lag4   -0.075675027 -0.06107462
## Lag5    1.000000000 -0.05851741
## Volume -0.058517414  1.00000000
```

Question b

```r
weekly = Weekly[,-1]
attach(weekly)
```

```r
set.seed(1)
rowTrain <- createDataPartition(y = weekly$Direction,
                                p = 0.75,
                                list = FALSE)
```

```r
glm.fit = glm(Direction ~ ., data = weekly, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ ., family = binomial, data = weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

The smallest p-value here is associated with Lag2. At the 5% level of significance, lag2 is the only predictor that is statistically significant.

Question c

```
test.pred.prob  <- predict(glm.fit, newdata = weekly, type = "response")

test.pred <- rep("Down", length(test.pred.prob))

test.pred[test.pred.prob > 0.5] <- "Up"
```

```
confusionMatrix(data = as.factor(test.pred),
                reference = weekly$Direction,
                positive = "Up")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down  Up
##       Down   54  48
##       Up    430 557
##
##                Accuracy : 0.5611
##                  95% CI : (0.531, 0.5908)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : 0.369
##
##                   Kappa : 0.035
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9207
##             Specificity : 0.1116
##          Pos Pred Value : 0.5643
##          Neg Pred Value : 0.5294
##              Prevalence : 0.5556
##          Detection Rate : 0.5115
##    Detection Prevalence : 0.9063
##       Balanced Accuracy : 0.5161
##
##        'Positive' Class : Up
##
```

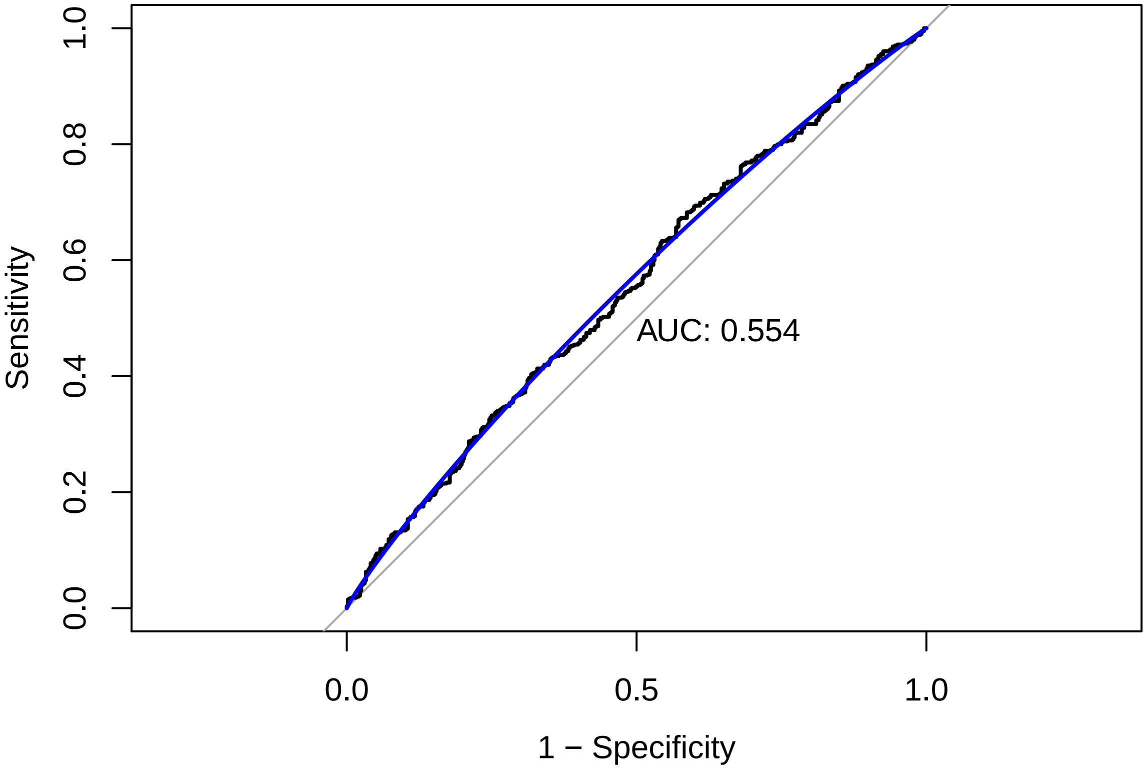
```
mean(test.pred == weekly$Direction)
```

```
## [1] 0.5610652
```

The diagonal elements of the confusion matrix indicate correct predictions, while the off-diagonals represent incorrect predictions. Hence our model correctly predicted that the market would go up on 124 weeks and that it would go down on 18 days, for a total of 54 + 557 = 611 correct predictions. We also see that the model predicts 56.1% of the time.

Question d

```
roc.glm <- roc(weekly$Direction, test.pred.prob)

plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



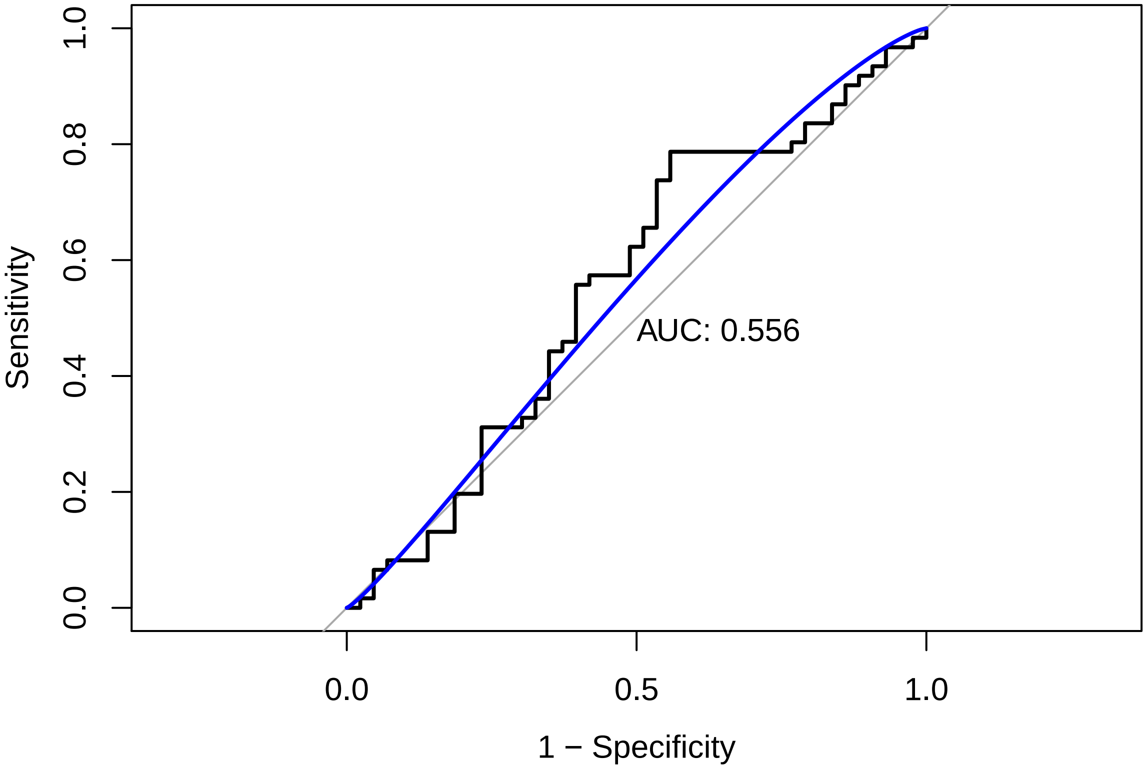
The AUC is 0.554.

Question e

```
train = (Weekly$Year < 2009)
weekly_2008 = weekly[!train,1:2]
Direction_2008 = weekly$Direction[!train]
```

```
glm.fit = glm(Direction ~ Lag1 + Lag2, data = weekly, family = binomial, subset = train)
glm.probs = predict(glm.fit, weekly_2008, type = "response")

test.pred <- rep("Down", length(glm.probs))
test.pred[glm.probs > 0.5] <- "Up"
```

```r
roc.glm <- roc(Direction_2008, glm.probs)
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```

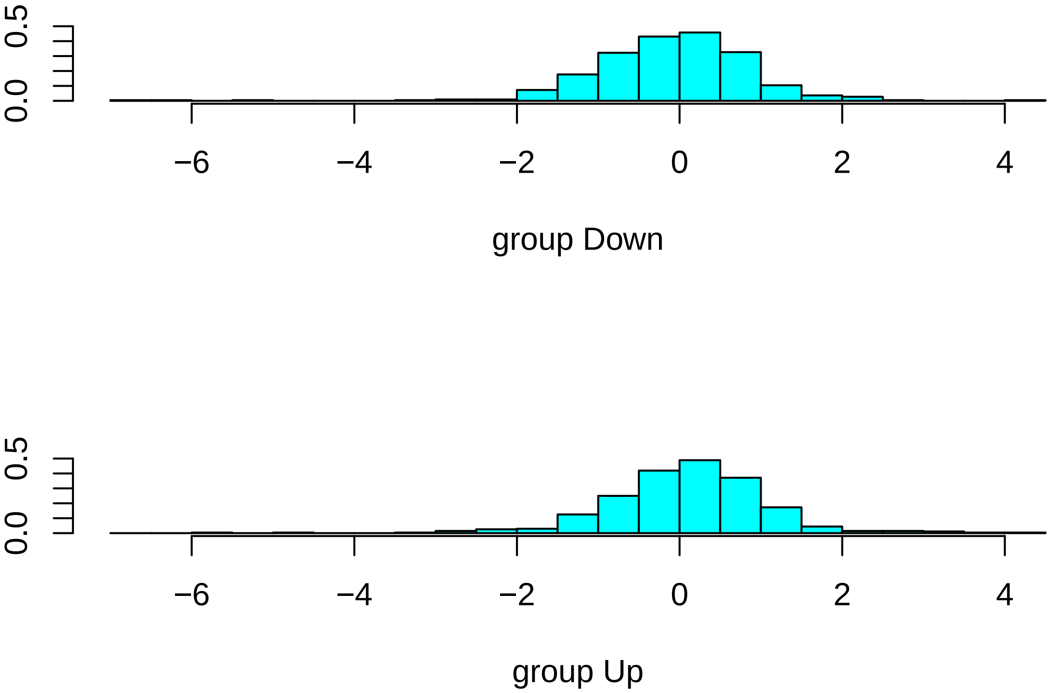


The AUC for the logistic regression model with just Lag1 and Lag2 is 0.557.

Question f

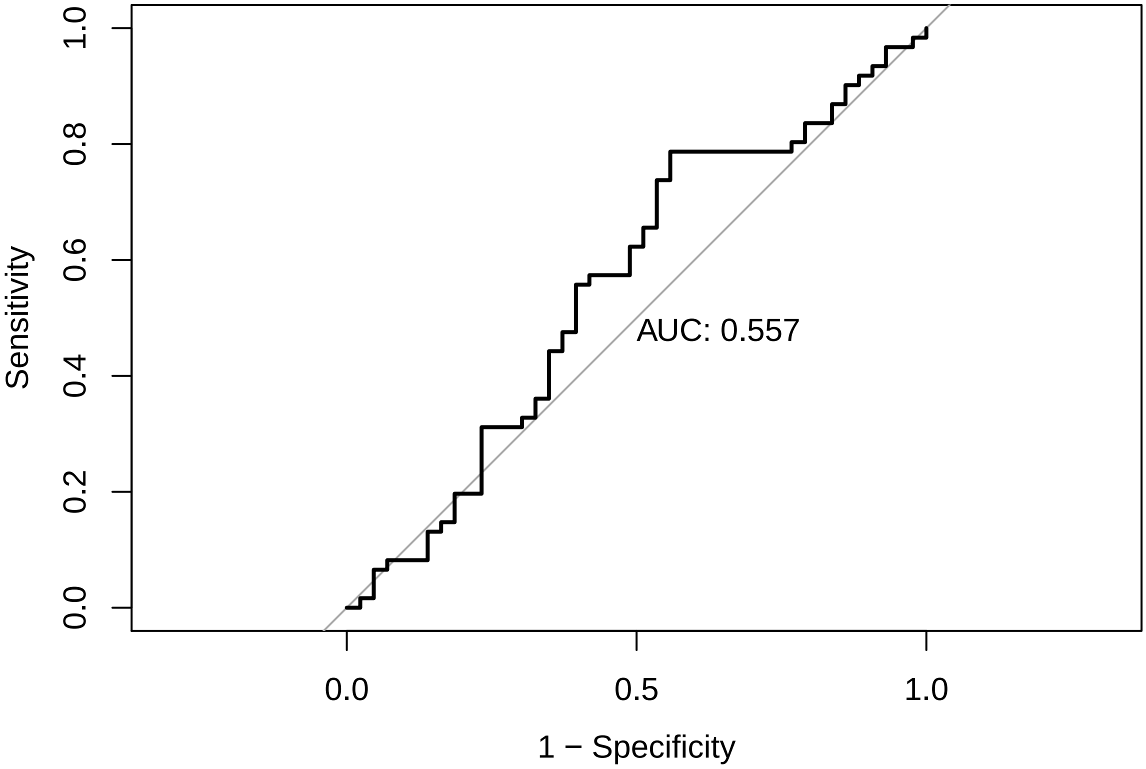
LDA

```r
lda.fit <- lda(Direction ~ Lag1 + Lag2, data = weekly, subset = train)
plot(lda.fit)
```



Evaluate the test set performance using ROC.

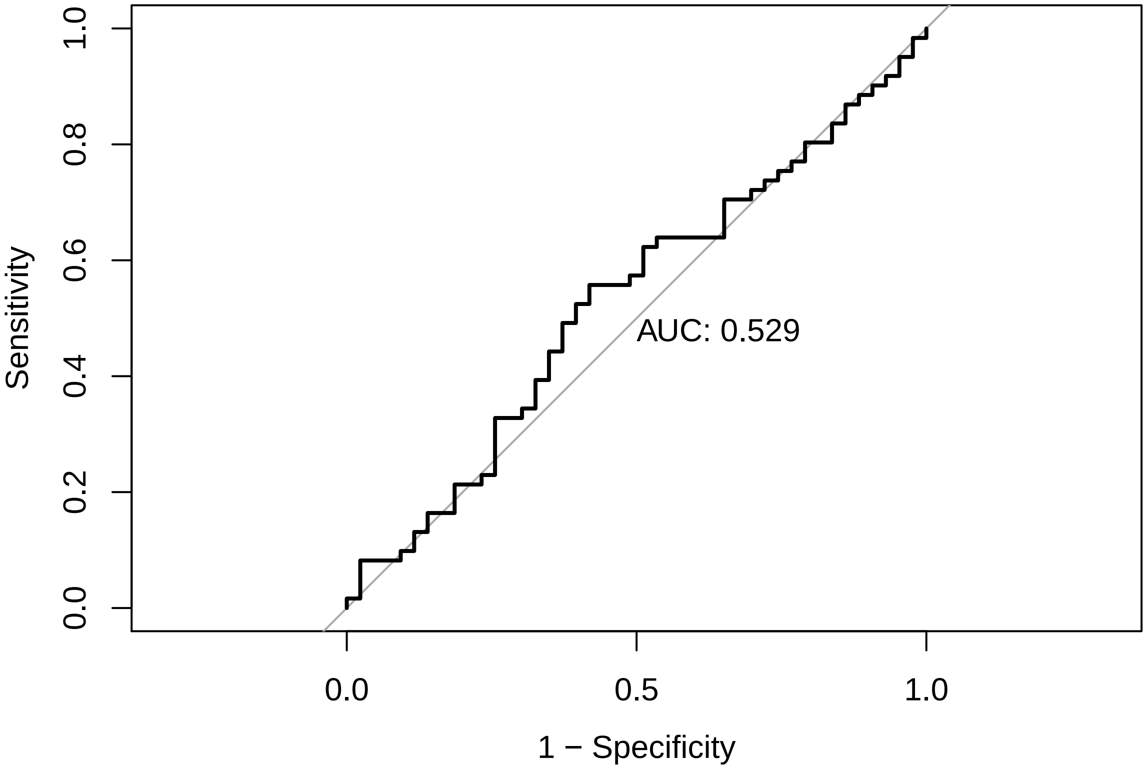
```
lda.pred <- predict(lda.fit, newdata = weekly_2008)

roc.lda <- roc(Direction_2008, lda.pred$posterior[,2],
               levels = c("Down", "Up"))
plot(roc.lda, legacy.axes = TRUE, print.auc = TRUE)
```



The AUC for LDA is 0.557.

QDA

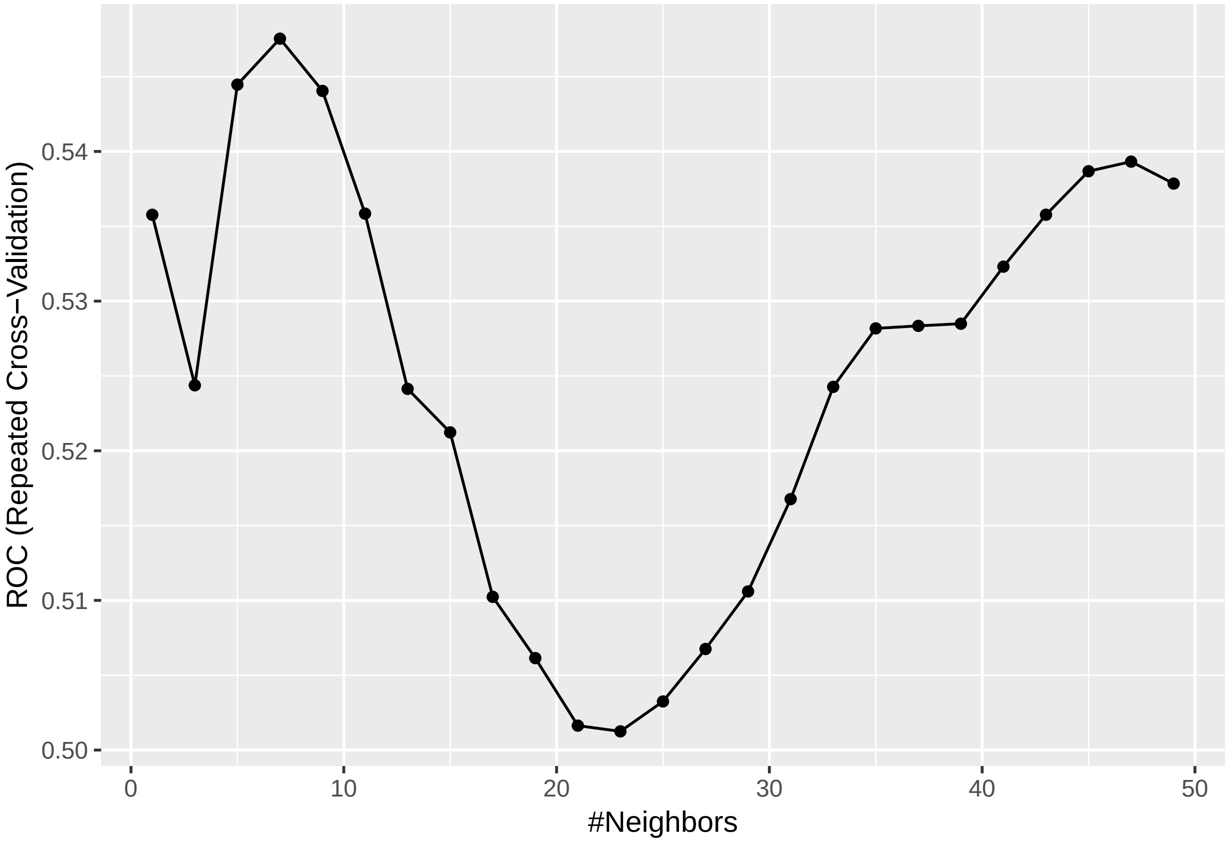
```
# use qda() in MASS
qda.fit <- qda(Direction ~ Lag1 + Lag2, data = weekly, subset = train)

qda.pred <- predict(qda.fit, newdata = weekly_2008)

roc.qda <- roc(Direction_2008, qda.pred$posterior[,2],
               levels = c("Down", "Up"))
plot(roc.qda, legacy.axes = TRUE, print.auc = TRUE)
```



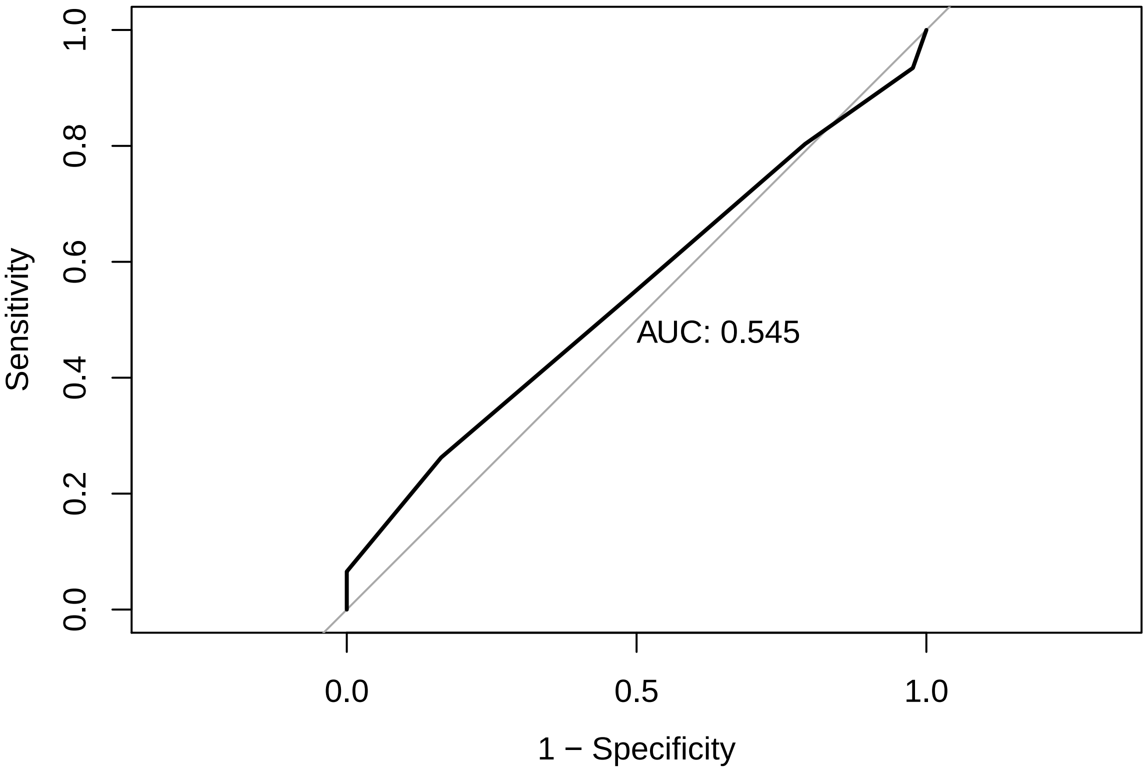
The AUC for QDA is 0.529.

Question g

```
ctrl <- trainControl(method = "repeatedcv",
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

set.seed(1)
model.knn <- train(x = weekly[train,1:2],
                   y = weekly$Direction[train],
                   method = "knn",
                   preProcess = c("center","scale"),
                   tuneGrid = data.frame(k = seq(1, 50, by = 2)),
                   trControl = ctrl,
                   metric = 'ROC')

ggplot(model.knn)
```



```
knn.pred <- predict(model.knn, newdata = weekly_2008, type = "prob")[,2]

roc.knn <- roc(Direction_2008, knn.pred)
plot(roc.knn, legacy.axes = TRUE, print.auc = TRUE)
```



AUC for the K Nearest Neighbor is 0.545.