Statistical Learning HW 1

Christian Conroy

February 10, 2018

1. Book #6 [4.7.6] (3 Points)

Suppose we collect data for a group of students in a statistics class with variables X1 = hours studied, X2 = undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, ??^0 = ???6, ??^1 = 0.05, ??^2 = 1.

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

```
exp(-6 + .05*40 + 1*3.5)/(1+exp(-6 + .05*40 + 1*3.5))
```

```
## [1] 0.378
```

(b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

```
# Hours = log(p/1-p) - ((B0 + B2*3.5)/B1)
log(.50/(1-.50)) - ((-6 + 3.5)/0.05)
```

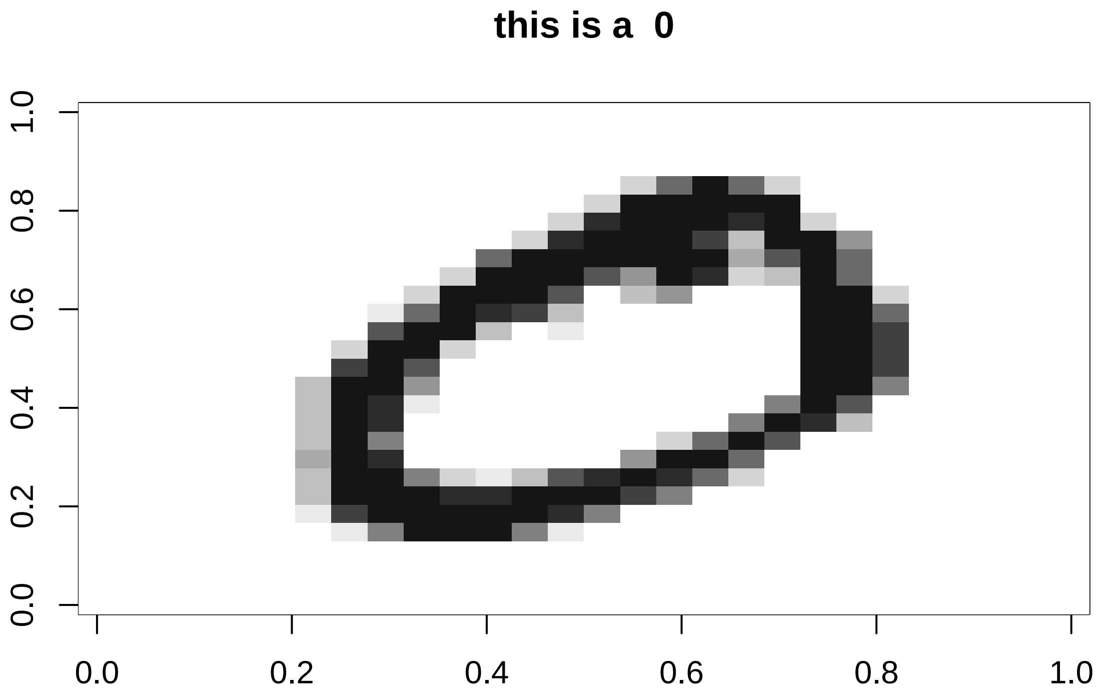
```
## [1] 50
```

- 2. Extra #25 (3 Points)

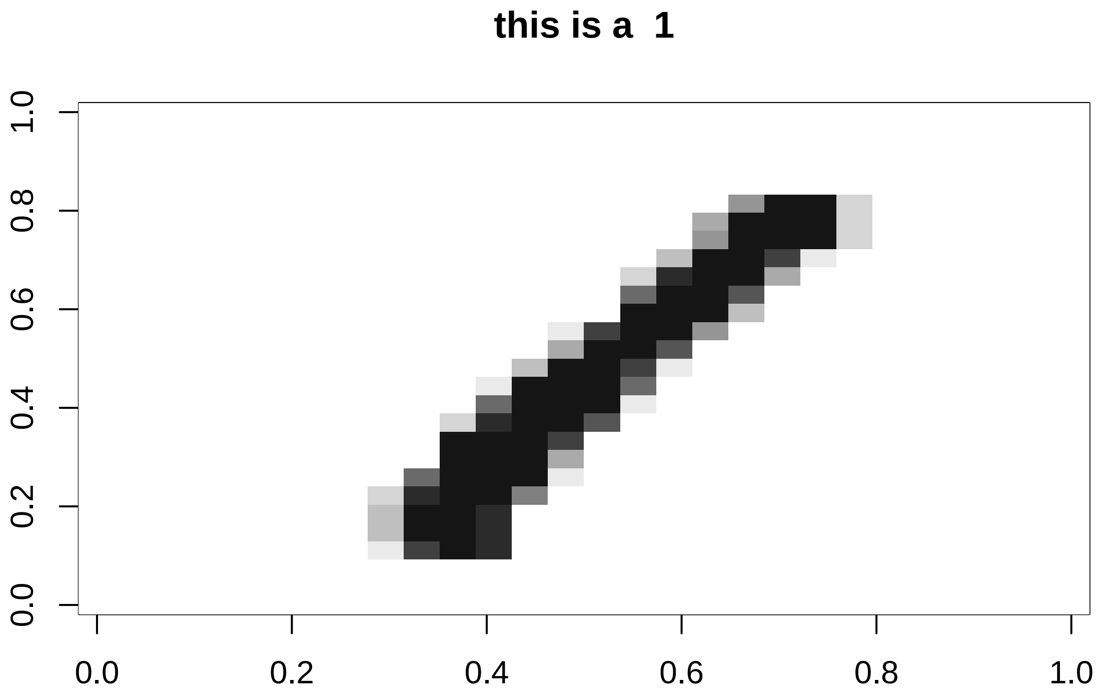
```
data <-load('mnist_all.RData')

plot_digit <- function(j){
arr784 <- as.numeric(train$x[j,])
col=gray(12:1/12)
image(matrix(arr784, nrow=28)[,28:1], col=col,
main = paste("this is a ",train$y[j]))
}
```

Familiarizing with the data

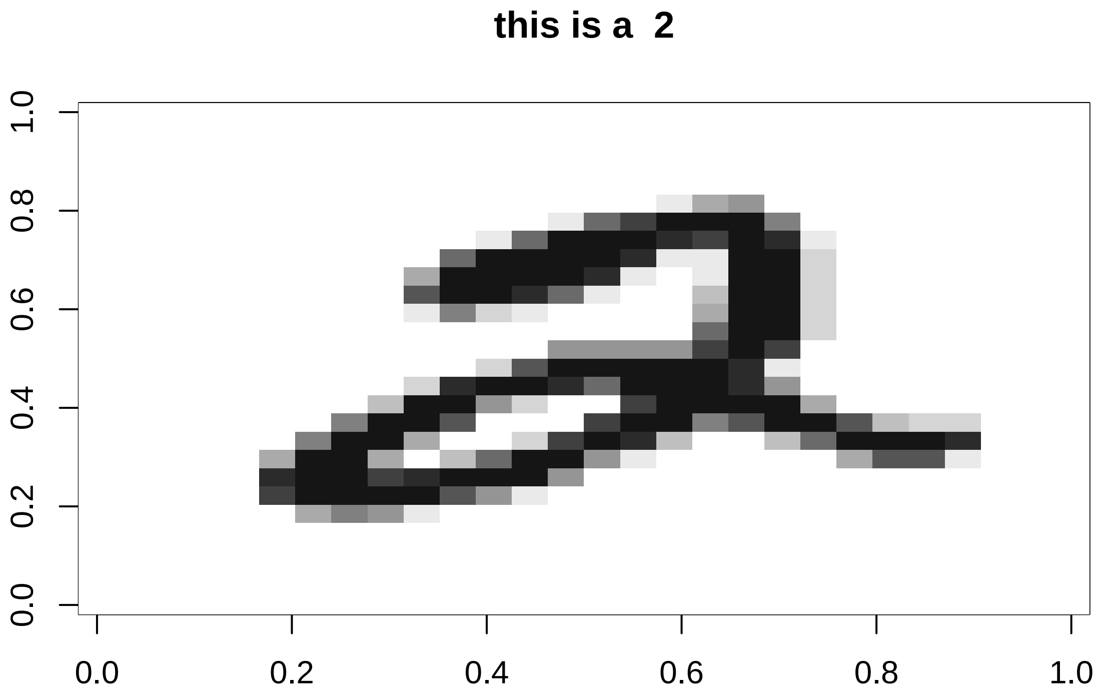
```
# Visualize digits 0-9
plot_digit(2) #0
```



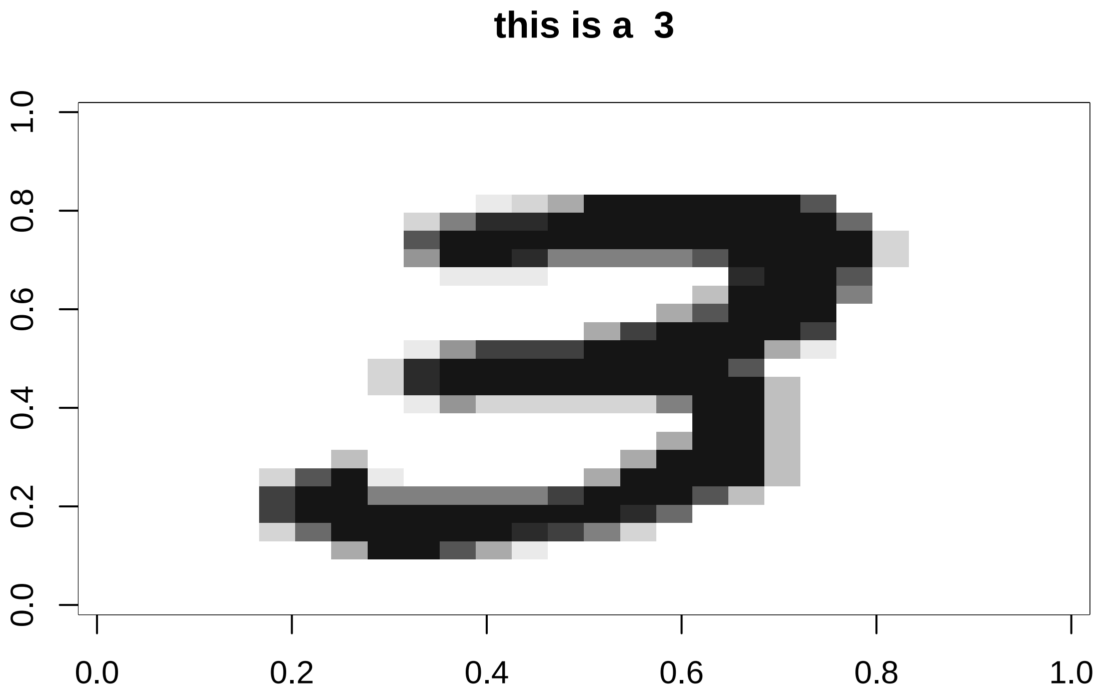
```
plot_digit(4) #1
```



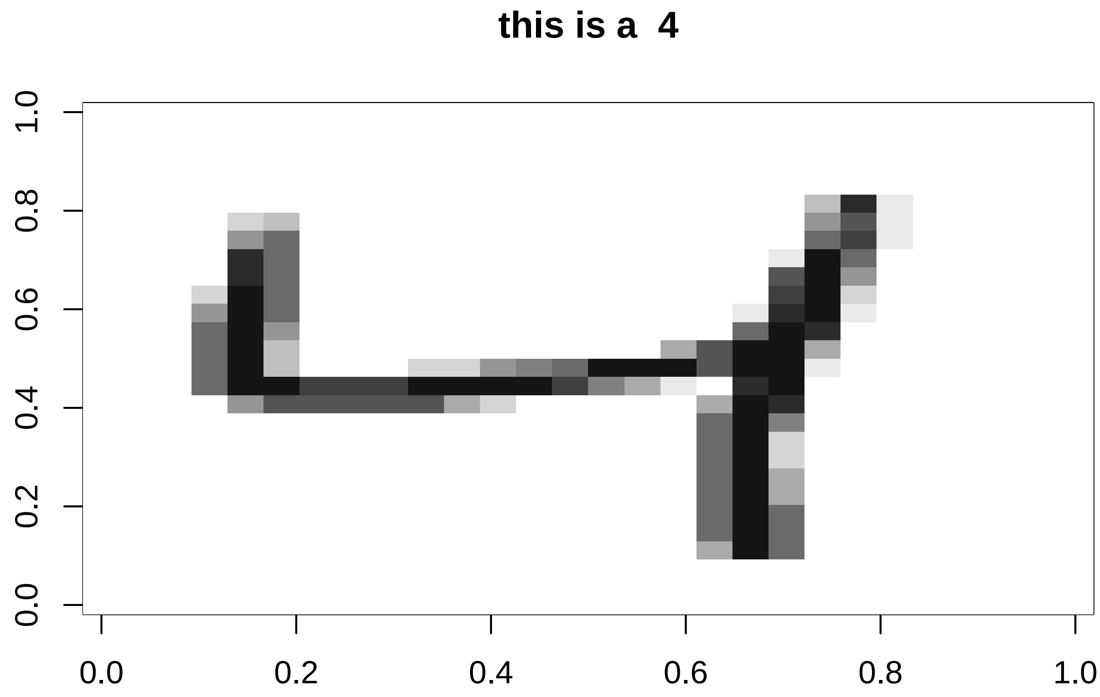
```
plot_digit(6) #2
```



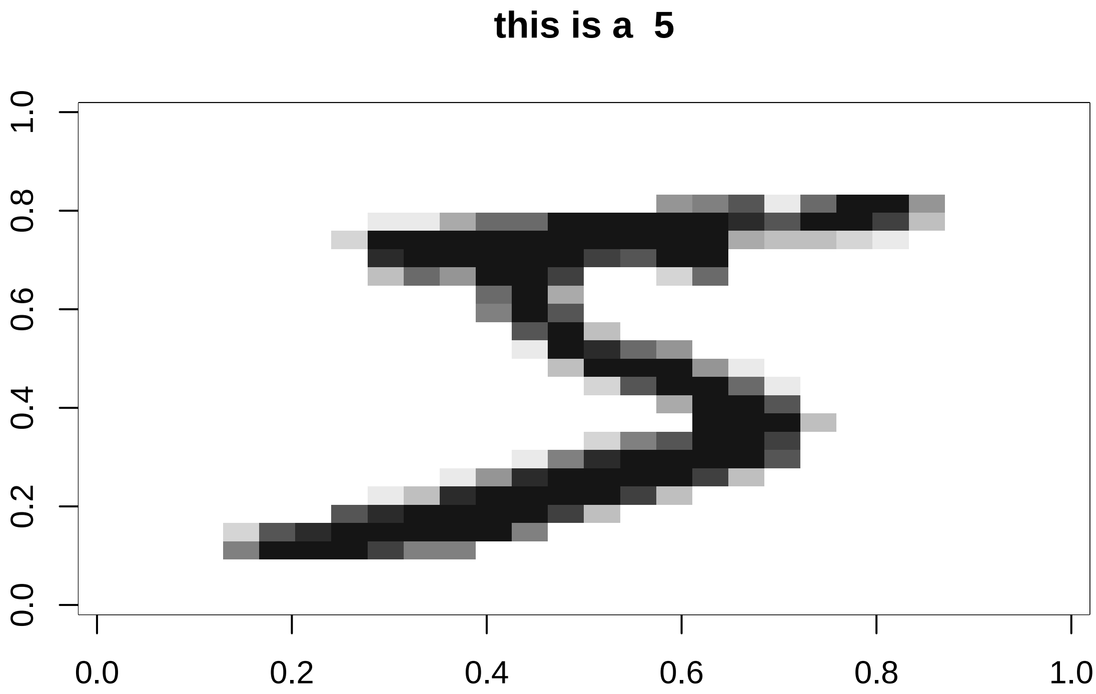
```
plot_digit(8) #3
```



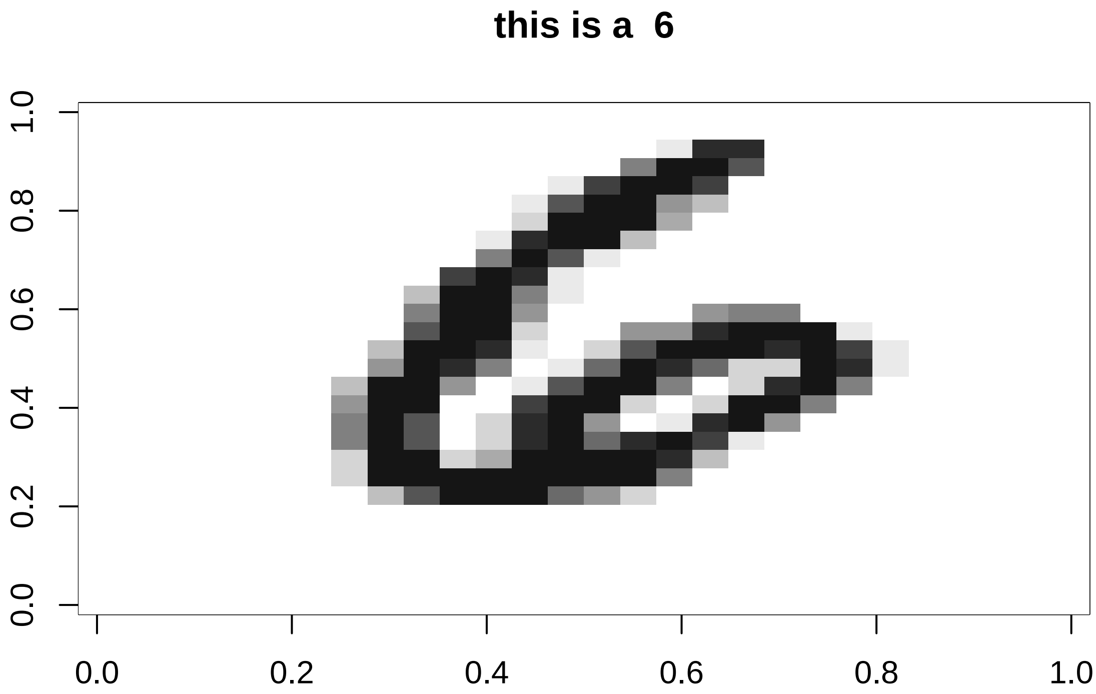
```
plot_digit(3) #4
```



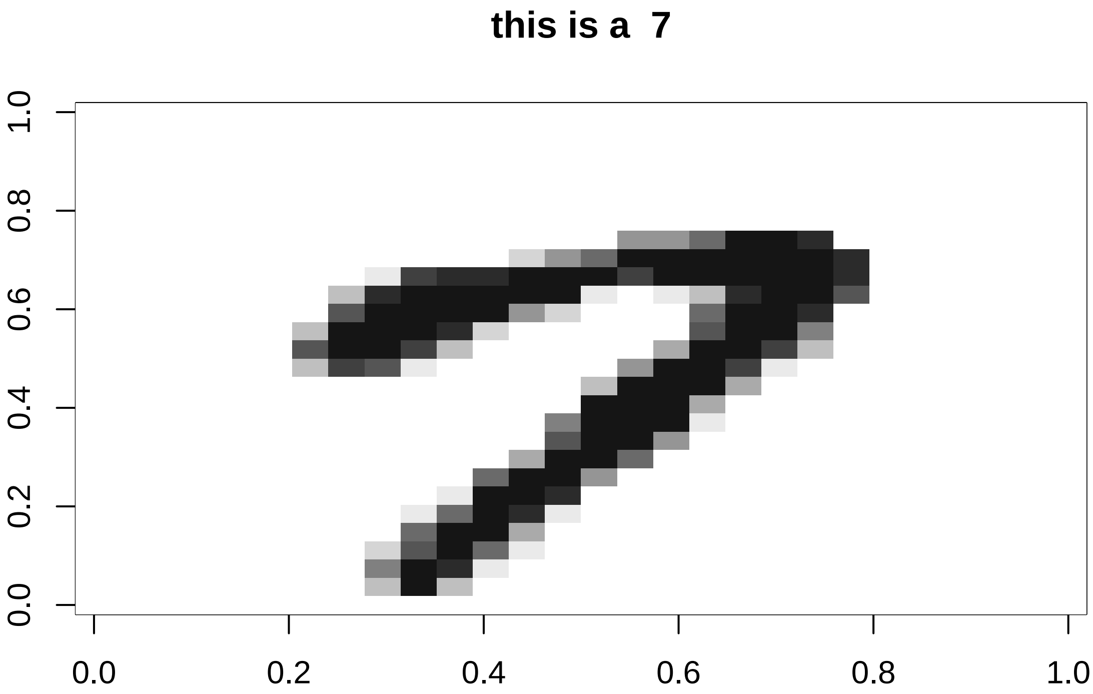
```
plot_digit(1) #5
```



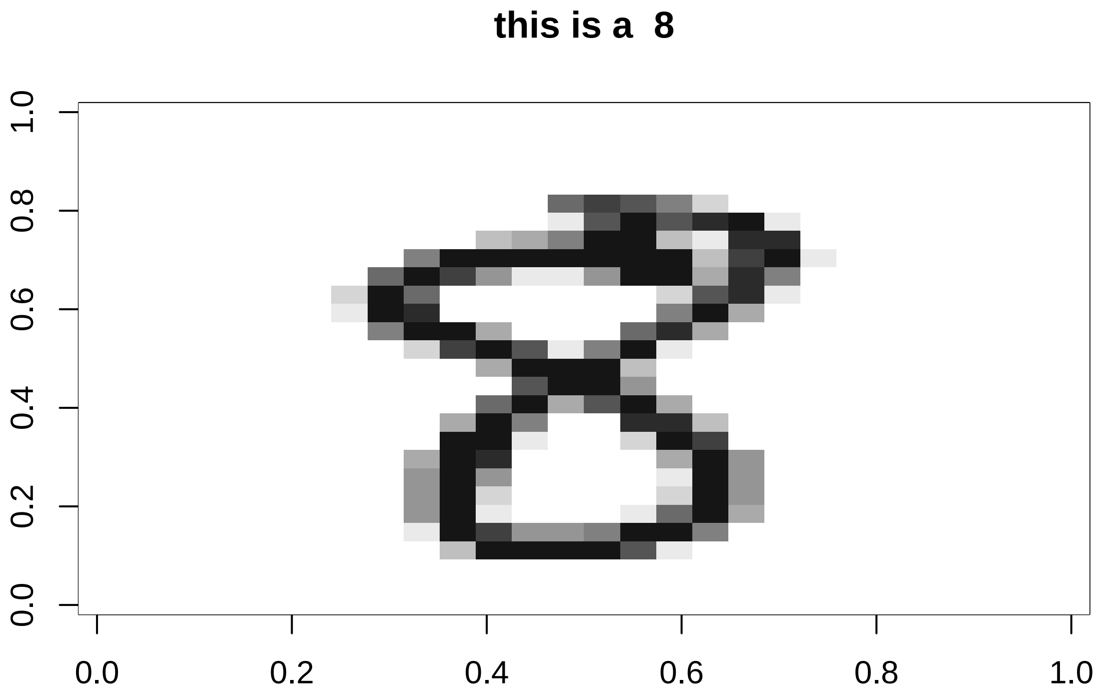
```r
plot_digit(14) #6
```



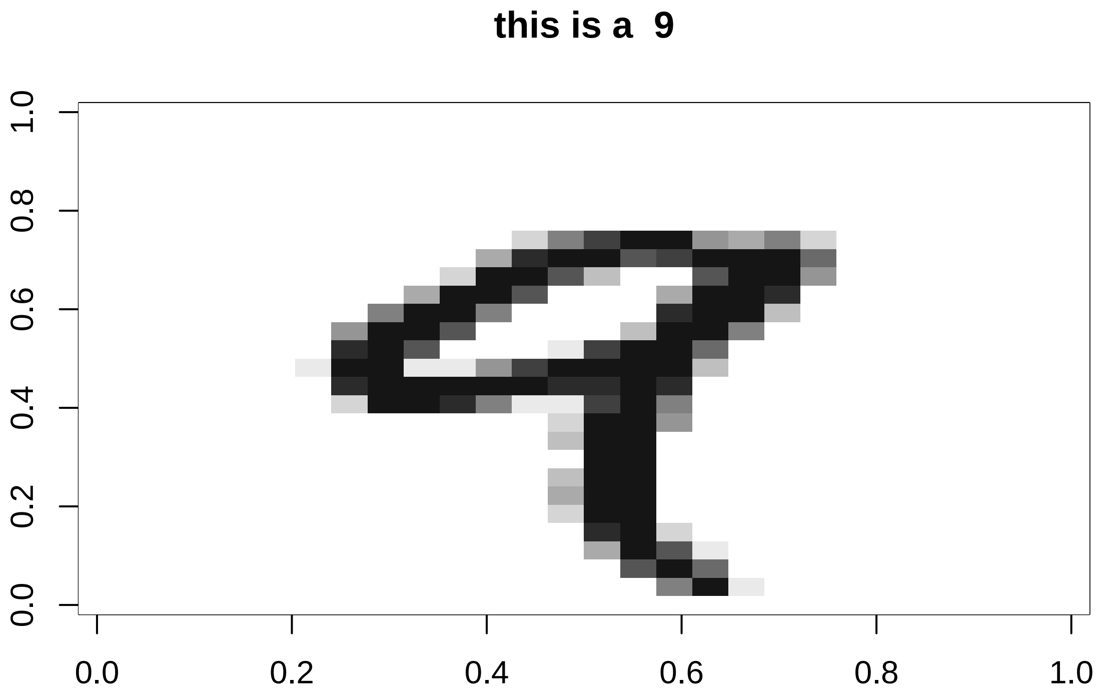
```
plot_digit(16) #7
```



```
plot_digit(47) #8
```



```r
plot_digit(5) #9
```



```
# find pixels that have zero variability
summary(train$x[,10]) # All X at this Y have 0 variability
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0
```

```
summary(train$x[1,100])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0
```

```
summary(train$x[900,400])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0
```

```
summary(train$x[,700]) # All X at this Y have 0 variability
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0
```

```
summary(train$x[,781]) # All X at this Y have 0 variability
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0
```

```
# find pixels that have positive variability
summary(train$x[,100])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0    13.2     0.0   255.0
```

```
summary(train$x[,200])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0     0.6     0.0   255.0
```

```
summary(train$x[,300])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0    15.0    94.2   231.0   255.0
```

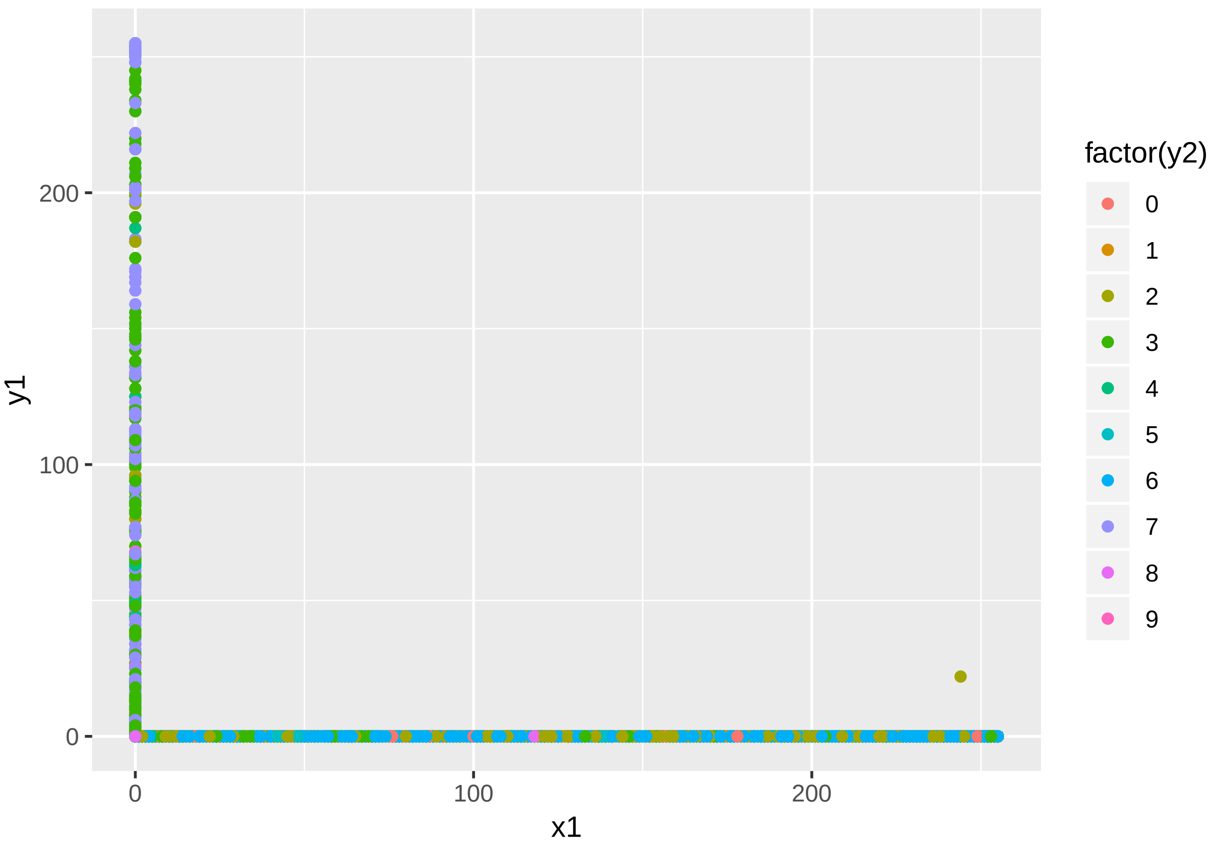
```
summary(train$x[,400])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0    50.5    47.0   255.0
```

```
summary(train$x[,500])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0    12.8     0.0   255.0
```

```
# Pick several pairs of features that both have non-zero variability. Make a scatterplot of these two
# features against each other. Label the datapoints with colors corresponding to the digits.
x1 <- train$x[,100]
y1 <- train$x[,200]
y2 <- train$y
forplot <- data.frame(x1, y1, y2)

ggplot(forplot, aes(x=x1, y=y1)) +geom_point(aes(col=factor(y2)))
```



Build a classifier, using only one variable (pixel). This variable should have large variation. Give the summary of the model and write out the logistic regression equation that has been obtained. Determine the fraction of true positives on the test set if the fraction of false positives on the training set is kept to 0.1.

```
# Prep Train Data
pixels <- data.frame(train$n, train$x, train$y)
names(pixels)[1] <- c("N")
names(pixels)[786] <- c("Digit")

sd(pixels$X350)
```

```
## [1] 103
```

```
# Looks to be a fairly large Variance

spec1 <- as.formula("factor(Digit) ~ X350")

fit <- glm(formula = spec1 , data = pixels, family=binomial)

pixels$pred = predict(fit, pixels, type = "response")
pixels$class_pred = predict(fit, pixels, type = "response") > .1

# We create a threshold at .1 and get a Boolean vector based on true prediction or false prediction

#accuracy - which rows did we get the class prediction correctly equal to z
sum(pixels$class_pred == pixels$Digit) / nrow(pixels)
```

```
## [1] 0.112
```

```r
# confusion matrix
table(pixels$Digit, pixels$class_pred)
```

```
##
##     TRUE
##   0 5923
##   1 6742
##   2 5958
##   3 6131
##   4 5842
##   5 5421
##   6 5918
##   7 6265
##   8 5851
##   9 5949
```

```r
# Use on the test data
test_pixels <- data.frame(test$n, test$x, test$y)
names(test_pixels)[1] <- c("N")
names(test_pixels)[786] <- c("Digit")

test_pixels$pred <- predict(fit, newdata = test_pixels, type = 'response')
test_pixels$class_pred = predict(fit, test_pixels, type = "response") > .1

sum(test_pixels$class_pred == test_pixels$Digit) / nrow(test_pixels)
```

```
## [1] 0.114
```

The fraction of true positives is 0.112 for the train set and 0.114 for the test set, which is not very good. Because I have a strict threshold, it is harder to get true positives of course.

- 3. Extra #26 (3 Points)

Choose two variables that have small correlation and large variation. Find the area under the ROC curve (auc) using the training data and the test data. Make a scatter plot of the two variables, colored by the type of digit, and use this to explain the performance of the classifier.

```r
sd(pixels$X350)
```

```
## [1] 103
```

```r
sd(pixels$X125)
```

```
## [1] 69.3
```

```r
cor(pixels$X350, pixels$X125)
```

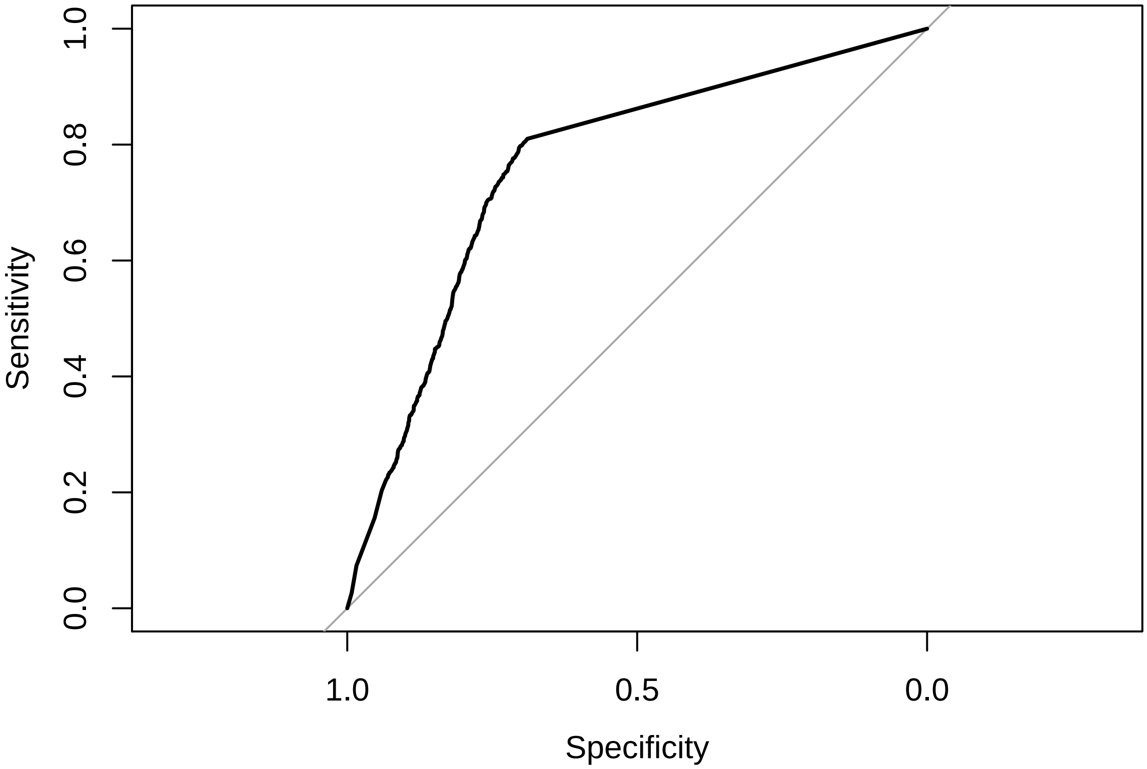
```
## [1] 0.0403
```

```r
# Looks to be decently large variation and small correlation.

spec2 <- as.formula("factor(Digit) ~ X350 + X125")

fit2 <- glm(formula = spec2, data = pixels, family=binomial)

test_pixels$pred <- predict(fit2, newdata = test_pixels, type = 'response')
```

In the above, I use pixels 350 and 125 because both respectively have fairly large variation and minimal correlation to each other.

ROC for the Train Data

```
# install.packages('pROC')
library(pROC)
r <- roc(pixels$Digit, pixels$pred)
plot(r)
```



```
auc(pixels$Digit, pixels$pred)
```

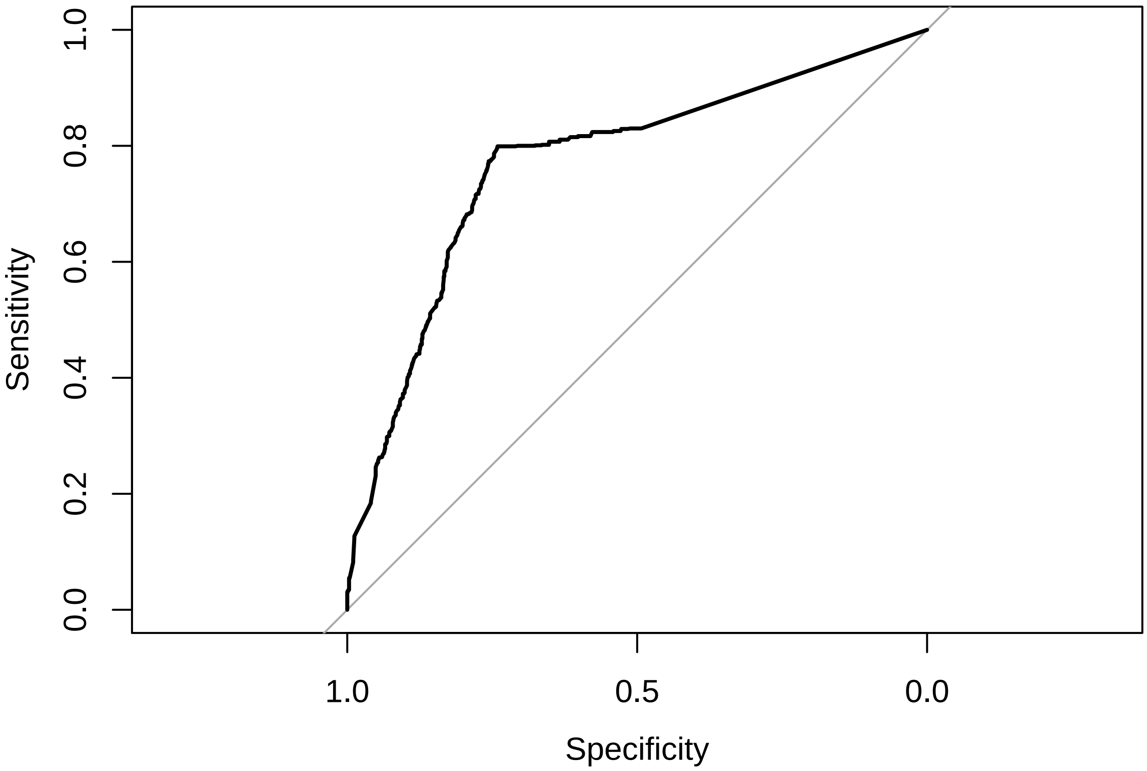
```
## Area under the curve: 0.763
```

```
# If you have an AOC less than 0.5, it means I'm essentially doing worse than chance and flipping my pr
```

The classification performance is not great, with an AOC of 0.763 that falls between an AOC of 0.5 which would indicate that flipping all predictions might be more effective and 1.0 which would indicate perfect classification. The ROC plot shows as much, with the curve somewhere halfway between a right angle and a diagonal.

ROC for the Test Data

```
r <- roc(test_pixels$Digit, test_pixels$pred)
plot(r)
```

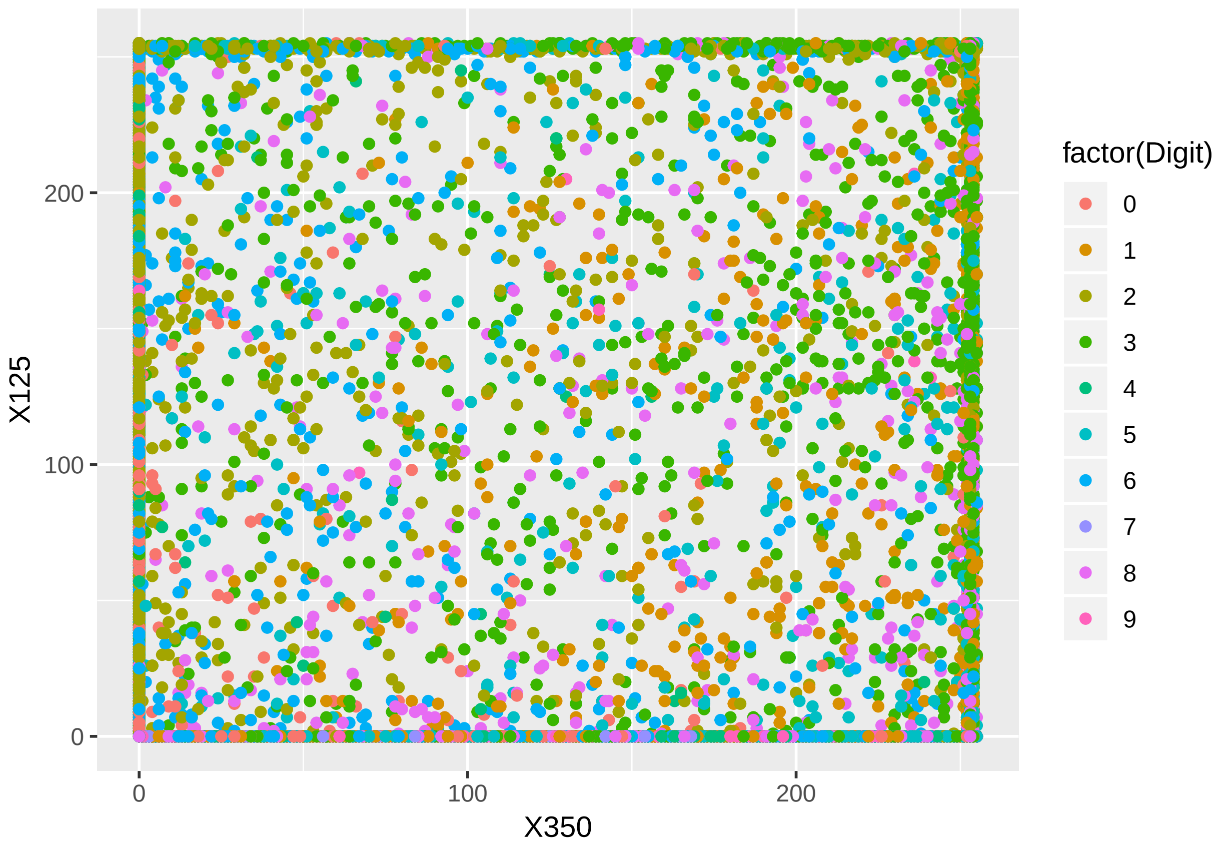


```
auc(test_pixels$Digit, test_pixels$pred)
```

```
## Area under the curve: 0.771
```

Because we have less information in the test set, the AOC is a bit lower than that of the train set, but both do fairly average at about halfway between 0.5 and 1. It makes sense that the AOC would be a bit lower on the test set as we don't have the same amount of information that we had on the train set.

```
ggplot(pixels, aes(x=X350, y=X125)) +geom_point(aes(col=factor(Digit)))
```



It's pretty clear from the above why it does such a bad job. There is no clear discernible pattern that can be identified in order to delineate between digits. As we saw earlier, there is low correlation between the two pixel variables.

- 4.

10a-d (5 points)

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

```
data("Weekly")
head(Weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484  0.155 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229  0.149 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936  0.160  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572  0.162  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816  0.154  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270  0.154 -1.372      Down
```

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

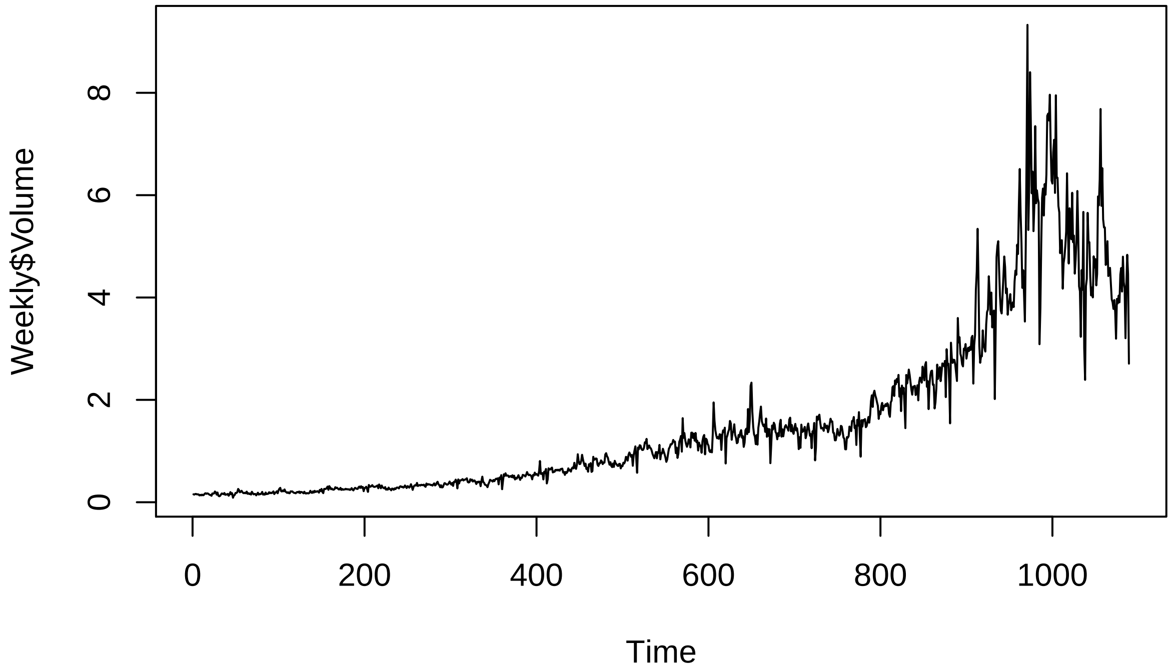
```r
# Numerical Summaries
summary(Weekly)
```

```
##       Year           Lag1             Lag2             Lag3        
##  Min.   :1990   Min.   :-18.20   Min.   :-18.20   Min.   :-18.20  
##  1st Qu.:1995   1st Qu.: -1.15   1st Qu.: -1.15   1st Qu.: -1.16  
##  Median :2000   Median :  0.24   Median :  0.24   Median :  0.24  
##  Mean   :2000   Mean   :  0.15   Mean   :  0.15   Mean   :  0.15  
##  3rd Qu.:2005   3rd Qu.:  1.40   3rd Qu.:  1.41   3rd Qu.:  1.41  
##  Max.   :2010   Max.   : 12.03   Max.   : 12.03   Max.   : 12.03  
##       Lag4             Lag5            Volume         Today        
##  Min.   :-18.20   Min.   :-18.20   Min.   :0.09   Min.   :-18.20  
##  1st Qu.: -1.16   1st Qu.: -1.17   1st Qu.:0.33   1st Qu.: -1.15  
##  Median :  0.24   Median :  0.23   Median :1.00   Median :  0.24  
##  Mean   :  0.15   Mean   :  0.14   Mean   :1.57   Mean   :  0.15  
##  3rd Qu.:  1.41   3rd Qu.:  1.40   3rd Qu.:2.05   3rd Qu.:  1.40  
##  Max.   : 12.03   Max.   : 12.03   Max.   :9.33   Max.   : 12.03  
##  Direction 
##  Down:484  
##  Up  :605  
##            
##            
##            
##            
```

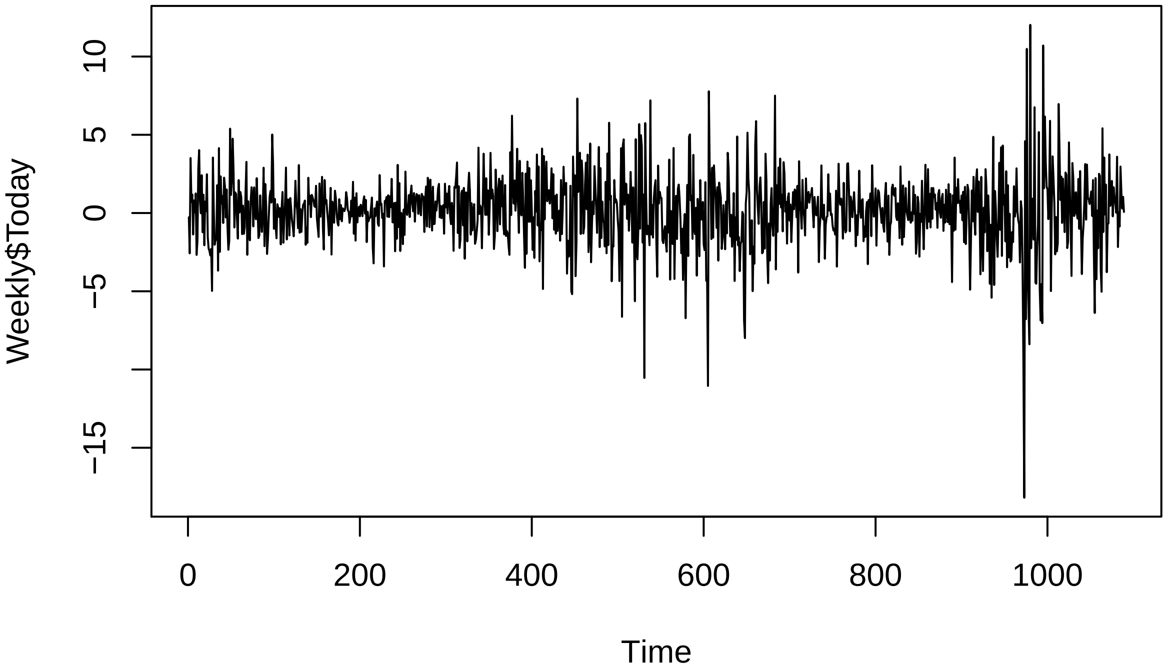
```r
cor(Weekly[,-9])
```

```
##           Year     Lag1    Lag2    Lag3      Lag4     Lag5  Volume    Today
## Year    1.0000 -0.03229 -0.0334 -0.0300 -0.03113 -0.03052  0.8419 -0.03246
## Lag1   -0.0323  1.00000 -0.0749  0.0586 -0.07127 -0.00818 -0.0650 -0.07503
## Lag2   -0.0334 -0.07485  1.0000 -0.0757  0.05838 -0.07250 -0.0855  0.05917
## Lag3   -0.0300  0.05864 -0.0757  1.0000 -0.07540  0.06066 -0.0693 -0.07124
## Lag4   -0.0311 -0.07127  0.0584 -0.0754  1.00000 -0.07568 -0.0611 -0.00783
## Lag5   -0.0305 -0.00818 -0.0725  0.0607 -0.07568  1.00000 -0.0585  0.01101
## Volume  0.8419 -0.06495 -0.0855 -0.0693 -0.06107 -0.05852  1.0000 -0.03308
## Today  -0.0325 -0.07503  0.0592 -0.0712 -0.00783  0.01101 -0.0331  1.00000
```

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



```
ts.plot(Weekly$Today)
```



In looking at the data, it is clear that the there is not a significant difference between the distribution of weekly returns across all lags. It looks as if the market changed more for the positive each week than it did negative. Interestingly, despite distributions being similar across the lagged variables, there is not strong correlation between the variables. It is also interesting that rather than simply becoming weaker in correlation as we lag further away, the correlation switches between negative and positive and decreases and increases in magnitude.

While the plot for volume shows that the average number of shares traded daily increased from 1990 to 2010, it is less pronounced at the end of the timeline, which is likely the result of the 2008 global financial crisis. The plot for today's price is interesting in that it seems to convey the old adage of passive investing that investing in the market through index funds is often a safe bet as price fluctuations always revert to the mean.

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
fit2 <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume , data = Weekly, family=binomial)
summary(fit2)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.695  -1.256   0.991   1.085   1.458
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2669     0.0859    3.11   0.0019 **
## Lag1         -0.0413     0.0264   -1.56   0.1181
## Lag2          0.0584     0.0269    2.18   0.0296 *
## Lag3         -0.0161     0.0267   -0.60   0.5469
## Lag4         -0.0278     0.0265   -1.05   0.2937
## Lag5         -0.0145     0.0264   -0.55   0.5833
## Volume       -0.0227     0.0369   -0.62   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
##
## Number of Fisher Scoring iterations: 4
```

Lag2 is statistically significant at the 95% confidence level.

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
Weekly$pred = predict(fit2, Weekly, type = "response")
Weekly$class_pred = predict(fit2, Weekly, type = "response") > .5

# We create a threshold at .5 and get a Boolean vector based on true prediction or false prediction

#accuracy - which rows did we get the class prediction correctly equal to z
sum(Weekly$class_pred == as.numeric(Weekly$Direction)) / nrow(Weekly)
```

```
## [1] 0.395
```

```
# WHY IS ABOVE GIVING ME 0??

# confusion matrix
table(Weekly$Direction, Weekly$class_pred)
```

```
##
##        FALSE TRUE
##   Down    54  430
##   Up      48  557
```

The classifer predicted correctly on only 39.5% of observations. The confusion matrix is telling me that there are two key mistakes that can be made by logistic regressions in classification. The first is a false positive, or in this case the number of observations predicted to be Up when they should have been predicted to be down. The confusion matrix shows that this happened for 48 cases. The second is a false negative, or in this case the number of observations predicted to be Down when they should have been predicted to be up. There are 54 of these cases. We use a decision threshold of 0.5. If we raise it to 0.6, meaning that we're comfortable with more false positives and negatives, the amount of cases in our confusion matrix increases.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```r
# Train on 1990 to 2008
Weekly_PreCrisis <- Weekly[Weekly$Year <= 2008,]

fit3 <- glm(Direction ~ Lag2, data = Weekly_PreCrisis, family=binomial)
summary(fit3)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly_PreCrisis)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -1.54   -1.26    1.02    1.09    1.37
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2033     0.0643    3.16   0.0016 **
## Lag2          0.0581     0.0287    2.02   0.0430 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1355
##
## Number of Fisher Scoring iterations: 4
```

```r
Weekly_PreCrisis$pred = predict(fit3, Weekly_PreCrisis, type = "response")
Weekly_PreCrisis$class_pred = predict(fit3, Weekly_PreCrisis, type = "response") > .5

# We create a threshold at .5 and get a Boolean vector based on true prediction or false prediction

#accuracy - which rows did we get the class prediction correctly equal to z
sum(Weekly_PreCrisis$class_pred == as.numeric(Weekly_PreCrisis$Direction)) / nrow(Weekly_PreCrisis)
```

```
## [1] 0.424
```

```r
# confusion matrix
table(Weekly_PreCrisis$Direction, Weekly_PreCrisis$class_pred)
```

```
##
##        FALSE TRUE
##   Down    23  418
##   Up      20  524
```

```r
# Test on 2009-2010
Weekly_PostCrisis <- Weekly[Weekly$Year>2008,]

Weekly_PostCrisis$pred <- predict(fit3, newdata = Weekly_PostCrisis, type = 'response')
Weekly_PostCrisis$class_pred = predict(fit3, Weekly_PostCrisis, type = "response") > .5

sum(Weekly_PostCrisis$class_pred == as.numeric(Weekly_PostCrisis$Direction)) / nrow(Weekly_PostCrisis)
```

```
## [1] 0.327
```

```
# confusion matrix
table(Weekly_PostCrisis$Direction, Weekly_PostCrisis$class_pred)
```

```
##
##        FALSE TRUE
##   Down     9   34
##   Up       5   56
```

When we restrict the data to the period of 1990 to 2008, we get an accuracy of 42.4% and false positives and negatives of 20 and 23 respectively using a decision threshold of 0.5. The accuracy goes down to 32.7% when we use the "test" dataset of post-2008 data. This makes sense as the stock prices differed drastically as a result of the global financial crisis and so a model trained on returns from 1990-2008 would not be very generalizable to a test set after the effects of the 2008 global financial crisis.

- 5. Extra #23 (5 points)

Replace the factor variable Purchase with a new numerical variable purchase01 which equals 1 if a customer bought Minute Maid orange juice (MM) and equals 0 if she bought Citrus Hill orange juice (CH).

```
data("OJ")
head(OJ)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            237       1    1.75    1.99   0.00    0.0         0
## 2       CH            239       1    1.75    1.99   0.00    0.3         0
## 3       CH            245       1    1.86    2.09   0.17    0.0         0
## 4       MM            227       1    1.69    1.69   0.00    0.0         0
## 5       CH            228       7    1.69    1.69   0.00    0.0         0
## 6       CH            230       7    1.69    1.99   0.00    0.0         0
##   SpecialMM LoyalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0   0.500        1.99        1.75      0.24     No     0.000
## 2         1   0.600        1.69        1.75     -0.06     No     0.151
## 3         0   0.680        2.09        1.69      0.40     No     0.000
## 4         0   0.400        1.69        1.69      0.00     No     0.000
## 5         0   0.957        1.69        1.69      0.00    Yes     0.000
## 6         1   0.965        1.99        1.69      0.30    Yes     0.000
##   PctDiscCH ListPriceDiff STORE
## 1    0.0000          0.24     1
## 2    0.0000          0.24     1
## 3    0.0914          0.23     1
## 4    0.0000          0.00     1
## 5    0.0000          0.00     0
## 6    0.0000          0.30     0
```

```
OJ$purchase01 <- ifelse((OJ$Purchase == "MM"), 1, 0)
OJ$Purchase <- NULL
```

a) Fit a logistic model to predict purchase01 from all predictors. Call this model fit.22a. There are several predictors for which a coefficient estimate is not available. Give a reason for each such predictor why this happens. Look for simple arithmetic relations between some of the predictors.

```
# Factor vars that need to  be factors
OJ$StoreID <- as.factor(OJ$StoreID)

# Change to glm
fit.22a <- glm(purchase01 ~ . , data = OJ, family = binomial)
summary(fit.22a)
```

```
##
## Call:
## glm(formula = purchase01 ~ ., family = binomial, data = OJ)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.752  -0.541  -0.231   0.526   2.800
##
## Coefficients: (6 not defined because of singularities)
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.9096     2.0816    2.36   0.0183 *
## WeekofPurchase  -0.0112     0.0114   -0.99   0.3232
## StoreID2        -0.0471     0.2907   -0.16   0.8712
## StoreID3        -0.2128     0.3907   -0.54   0.5859
## StoreID4        -0.5316     0.4191   -1.27   0.2046
## StoreID7        -0.6479     0.2881   -2.25   0.0245 *
## PriceCH          4.5649     1.8874    2.42   0.0156 *
## PriceMM         -3.6849     0.9155   -4.03  5.7e-05 ***
## DiscCH          11.2717    18.8239    0.60   0.5493
## DiscMM          25.4431     9.3129    2.73   0.0063 **
## SpecialCH        0.2634     0.3430    0.77   0.4427
## SpecialMM        0.3135     0.2759    1.14   0.2558
## LoyalCH         -6.2487     0.4104  -15.23  < 2e-16 ***
## SalePriceMM          NA         NA      NA       NA
## SalePriceCH          NA         NA      NA       NA
## PriceDiff            NA         NA      NA       NA
## Store7Yes            NA         NA      NA       NA
## PctDiscMM      -48.5569    19.5007   -2.49   0.0128 *
## PctDiscCH      -28.2608    35.5752   -0.79   0.4270
## ListPriceDiff        NA         NA      NA       NA
## STORE                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.85  on 1069  degrees of freedom
## Residual deviance:  816.25  on 1055  degrees of freedom
## AIC: 846.2
##
## Number of Fisher Scoring iterations: 5
```

```
alias(fit.22a)
```

```
## Model :
## purchase01 ~ WeekofPurchase + StoreID + PriceCH + PriceMM + DiscCH +
##     DiscMM + SpecialCH + SpecialMM + LoyalCH + SalePriceMM +
##     SalePriceCH + PriceDiff + Store7 + PctDiscMM + PctDiscCH +
##     ListPriceDiff + STORE
##
## Complete :
##               (Intercept) WeekofPurchase StoreID2 StoreID3 StoreID4
## SalePriceMM    0           0              0        0        0
## SalePriceCH    0           0              0        0        0
## PriceDiff      0           0              0        0        0
## Store7Yes      0           0              0        0        0
## ListPriceDiff  0           0              0        0        0
## STORE          1           0              1        2        3
##               StoreID7 PriceCH PriceMM DiscCH DiscMM SpecialCH SpecialMM
## SalePriceMM    0        0       1       0     -1       0         0
## SalePriceCH    0        1       0      -1      0       0         0
## PriceDiff      0       -1       1       1     -1       0         0
## Store7Yes      1        0       0       0      0       0         0
## ListPriceDiff  0       -1       1       0      0       0         0
## STORE         -1        0       0       0      0       0         0
##               LoyalCH PctDiscMM PctDiscCH
## SalePriceMM    0       0         0
## SalePriceCH    0       0         0
## PriceDiff      0       0         0
## Store7Yes      0       0         0
## ListPriceDiff  0       0         0
## STORE          0       0         0
```

A coefficient estimate is not available for SalePriceMM, SalePriceCH, PriceDiff, Store7, ListPriceDiff, and STORE. This is due to strong correlation between independent variables (i.e. collinearity). In fact, as the alias analysis above shows, there are several variables that are perfectly collinear.

b) Remove all predictors for which a coefficient estimate is not available and fit a new model. Call this model fit.22b. What are the differences between fit.22a and fit.22b, if any?

```r
fit.22b <- glm(purchase01 ~ . , data = OJ[,-c(10:13, 16:17)], family = binomial)
summary(fit.22b)
```

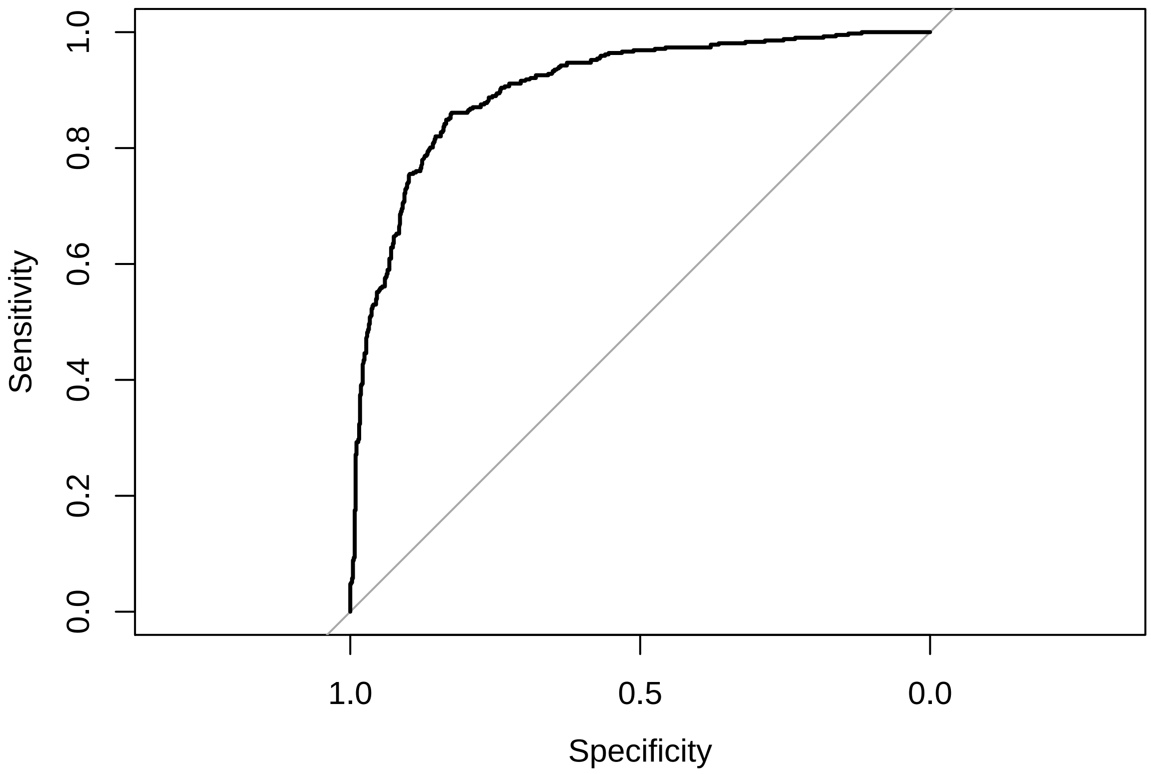
```
## 
## Call:
## glm(formula = purchase01 ~ ., family = binomial, data = OJ[, 
##     -c(10:13, 16:17)])
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
## -2.752  -0.541  -0.231   0.526   2.800  
## 
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       4.9096     2.0816    2.36   0.0183 *  
## WeekofPurchase   -0.0112     0.0114   -0.99   0.3232    
## StoreID2         -0.0471     0.2907   -0.16   0.8712    
## StoreID3         -0.2128     0.3907   -0.54   0.5859    
## StoreID4         -0.5316     0.4191   -1.27   0.2046    
## StoreID7         -0.6479     0.2881   -2.25   0.0245 *  
## PriceCH           4.5649     1.8874    2.42   0.0156 *  
## PriceMM          -3.6849     0.9155   -4.03  5.7e-05 ***
## DiscCH           11.2717    18.8239    0.60   0.5493    
## DiscMM           25.4431     9.3129    2.73   0.0063 ** 
## SpecialCH         0.2634     0.3430    0.77   0.4427    
## SpecialMM         0.3135     0.2759    1.14   0.2558    
## LoyalCH          -6.2487     0.4104  -15.23  < 2e-16 ***
## PctDiscMM       -48.5569    19.5007   -2.49   0.0128 *  
## PctDiscCH       -28.2608    35.5752   -0.79   0.4270    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.85  on 1069  degrees of freedom
## Residual deviance:  816.25  on 1055  degrees of freedom
## AIC: 846.2
##
## Number of Fisher Scoring iterations: 5
```

Now that the coefficients that could not be estimated due to perfect collinearity are out of the model, we notice that there is no change in the logistic regression summary, with the coefficients, standard errors, t statistics, and p-values remaining the same across models.

c) Which predictors are significant for fit.22b? Make a new model with only those predictors and call it fit.22c. Plot the ROC curve and show that the area under the curve is approximately 0.89.

```
OJ$pred = predict(fit.22b, OJ, type = "response")

r <- roc(OJ$purchase01, OJ$pred)
plot(r)
```



```
auc(OJ$purchase01, OJ$pred)
```

```
## Area under the curve: 0.904
```

StoreID7, PriceCH, PriceMM, DisMM, LoyalCH, and PctDiscCH are all statistically significant at either the 95% or 99% confidence levels.

AOC is 0.904, or approximately 0.89.

d) Consider now the predicted odds that a customer purchases Minute Maid. How do these odds change if the price of Minute Maid is decreased by $0.01? How do these odds change if the price of Citrus Hill is increased by $0.01? How do these odds change if the discount offered for Minute Maid is increased by $0.01? Note that this is essentially the same as dropping the price for Minute Maid, but the predicted effect on the odds is very different.

```
PriceMM <- OJ
PriceMM$PriceMM <- PriceMM$PriceMM -.01
PriceCH <- OJ
PriceCH$PriceCH <- PriceCH$PriceCH + .01
DiscMM <- OJ
DiscMM$DiscMM <- DiscMM$DiscMM +.01

# Price MM Decline
P1 <- predict(fit.22b, PriceMM, type = "response")
P0 <- predict(fit.22b, OJ, type = "response")

mean(P1 - P0, na.rm = TRUE)
```

```
## [1] 0.00444
```

```
# Price CH Increase
P1 <- predict(fit.22b, PriceCH, type = "response")
P0 <- predict(fit.22b, OJ, type = "response")

mean(P1 - P0, na.rm = TRUE)
```

```
## [1] 0.00551
```

```
# Disc MM Increase
P1 <- predict(fit.22b, DiscMM, type = "response")
P0 <- predict(fit.22b, OJ, type = "response")

mean(P1 - P0, na.rm = TRUE)
```

```
## [1] 0.031
```

The odds change by 0.0444 if the price of Minute Maid is decreased by 0.01. The odds change by 0.00551 if the price of Citrus Hill increases by 0.01. The odds change by 0.031 if the discount offered by Minute Maid increases by 0.01. The conclusion then is that the discount offers the best change of persuading more customers to buy Minute Maid.

- 6. Extra#27 (5 points)

Build a classifier that uses the 10 variables with the largest variances. Make ROC curves for training and test data and comment on the performance of the classifier. Is this a good way to select 10 predictors for classification? Can you think of other ways of selecting 10 predictors for classification?

```
# Get variables with the largest variance
all.sd <- apply(pixels, 2, sd)
head(sort(all.sd, decreasing=TRUE), 10)
```

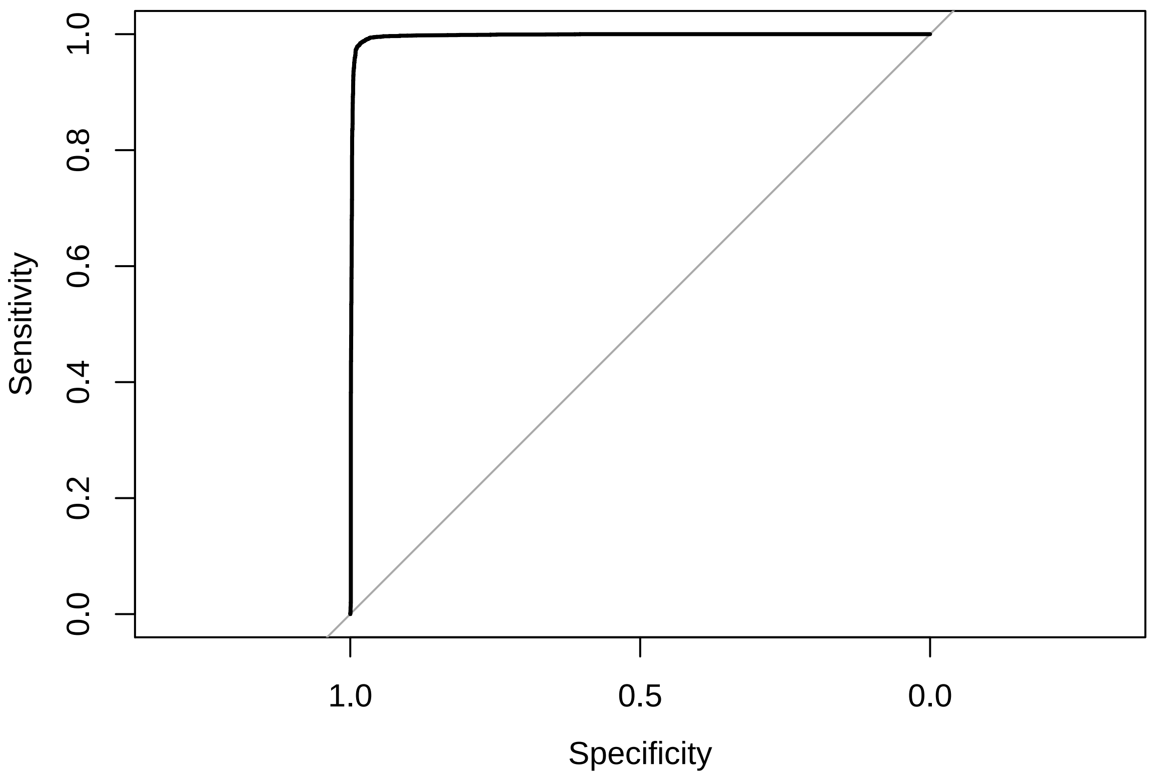
```
## X379 X407 X462 X628 X463 X435 X438 X434 X629 X410
##  114  114  113  113  113  113  113  113  112  112
```

```
# Build classified using those variables
fit6 <- glm(factor(Digit) ~ X379 + X407 + X462 + X628 + X463 + X435 + X438 + X434 + X629 + X410 , data

# Predict using train data
```

```
pixels$pred = predict(fit6, pixels, type = "response")

# Build the ROC Curve and Get AOC
r <- roc(pixels$Digit, pixels$pred)
```

```
## Warning in roc.default(pixels$Digit, pixels$pred): 'response' has more than
## two levels. Consider setting 'levels' explicitly or using 'multiclass.roc'
## instead
```

```
plot(r)
```



```
auc(pixels$Digit, pixels$pred)
```

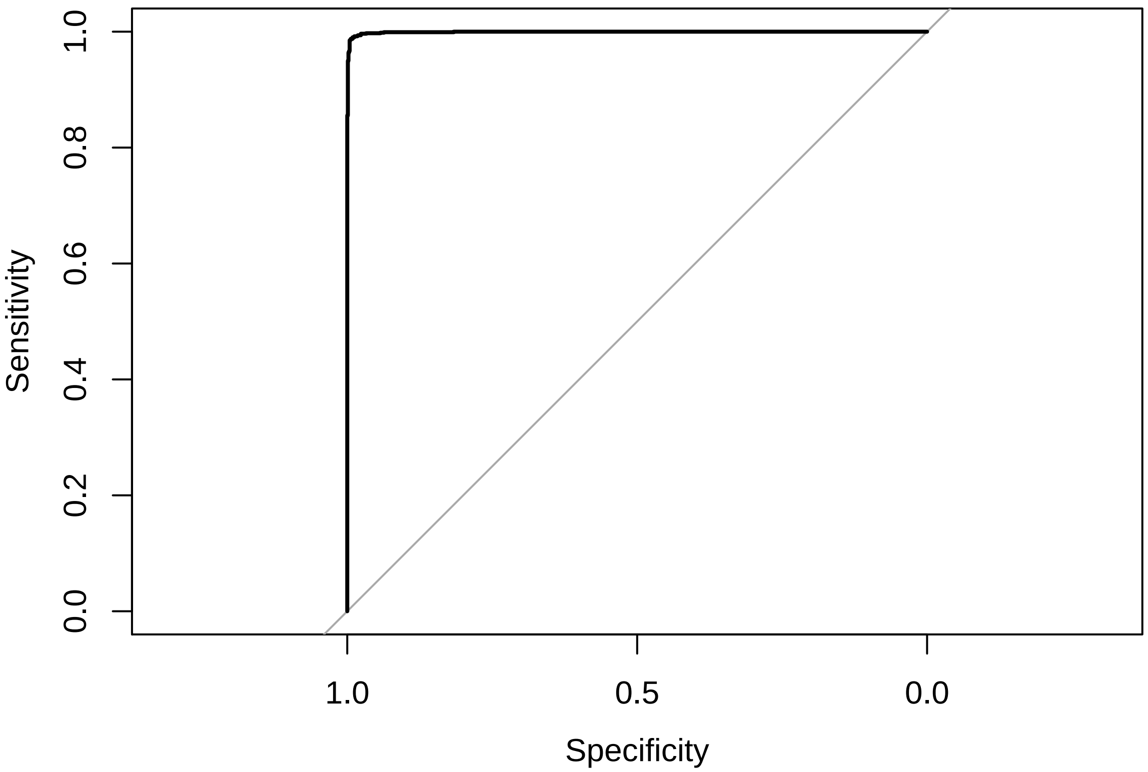
```
## Warning in roc.default(response, predictor, auc = TRUE, ...): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```

```
## Area under the curve: 0.997
```

```
# Predict using test data
test_pixels$pred <- predict(fit6, newdata = test_pixels, type = 'response')

# Build the ROC Curve and Get AOC
r <- roc(test_pixels$Digit, test_pixels$pred)
```

```
## Warning in roc.default(test_pixels$Digit, test_pixels$pred): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```

```
plot(r)
```



```
auc(test_pixels$Digit, test_pixels$pred)
```

```
## Warning in roc.default(response, predictor, auc = TRUE, ...): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```

```
## Area under the curve: 0.999
```

The predictors do a very good job at classification, as indicating by the fact that the ROC curves are both near right degree angles and the associated AOC are both very lose to 1. While this is probably the best way of selecting predictors for classification, I could also imagine using principal component analysis for classification. The idea is that we would look at all of the predictors and find combined components that explain the highest percentage in the variance of the predicted probability of being a digit.