HW4

Yigao Li

February 13, 2018

4 - 6

(a)

$$\text{P}\{\text{Y=receive an A}\} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\hat{\beta}_2 X_2}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\hat{\beta}_2 X_2}} = \frac{e^{-6+0.05X_1+X_2}}{1+e^{-6+0.05X_1+X_2}}$$

$$\text{P}\{\text{Y}|X_1=40, X_2=3.5\} = \frac{e^{-6+0.05\times 40+3.5}}{1+e^{-6+0.05\times 40+3.5}} \approx 0.37754$$

(b)

$$0.5 = \frac{e^{-6+0.05X_1+3.5}}{1+e^{-6+0.05X_1+3.5}}$$
$$X_1 = 50$$

4 - 13

```
library(MASS)
boston <- Boston
crim.median <- median(boston$crim)
boston$crimclass <- as.numeric(boston$crim > crim.median)
boston <- subset(boston, select = -crim)
splt <- floor(dim(boston)[1]*0.75)
train <- 1:splt
test <- (splt+1):dim(boston)[1]
boston.train <- boston[train,]
boston.test <- boston[test,]
crim.test <- boston$crimclass[test]
model.1 <- glm(crimclass ~ ., data = boston, family = binomial, subset = train)
summary(model.1)
```

```
##
## Call:
## glm(formula = crimclass ~ ., family = binomial, data = boston,
##     subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3643  -0.2584  -0.0315   0.1406   3.4545
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -42.799118   7.890475  -5.424 5.82e-08 ***
```

```
## zn           -0.066836   0.036975  -1.808 0.070668 .
## indus        -0.088478   0.051755  -1.710 0.087346 .
## chas          1.023592   0.753010   1.359 0.174041
## nox          59.170886   9.555240   6.193 5.92e-10 ***
## rm           -0.676176   0.816074  -0.829 0.407347
## age           0.008651   0.012974   0.667 0.504905
## dis           0.654216   0.232571   2.813 0.004909 **
## rad           0.621347   0.183638   3.384 0.000716 ***
## tax          -0.001433   0.003760  -0.381 0.703168
## ptratio       0.485265   0.141215   3.436 0.000590 ***
## black        -0.009549   0.006112  -1.562 0.118195
## lstat         0.068709   0.054148   1.269 0.204474
## medv          0.202732   0.080266   2.526 0.011546 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.3  on 378  degrees of freedom
## Residual deviance: 182.5  on 365  degrees of freedom
## AIC: 210.5
##
## Number of Fisher Scoring iterations: 8
```

```
prob.1 <- predict(model.1, boston.test, type = "response")
pred.1 <- rep(0, length(prob.1))
pred.1[prob.1 > 0.5] = 1
mean(pred.1 != crim.test)
```

```
## [1] 0.07874016
```

```
library(bestglm)
```

```
## Warning: package 'bestglm' was built under R version 3.4.3
```

```
## Loading required package: leaps
```

```
## Warning: package 'leaps' was built under R version 3.4.3
```

```
boston.for.bestglm <- within(boston, {
  y <- crimclass
  crimclass <- NULL
})
res.bestglm <- bestglm(Xy = boston.for.bestglm, family = binomial, IC = "AIC", method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
```

```
summary(res.bestglm$BestModel)
```

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4197  -0.1840  -0.0004   0.0022   3.4087
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -31.441272   6.048989  -5.198 2.02e-07 ***
## zn           -0.082567   0.031424  -2.628  0.00860 **
## nox          43.195824   6.452812   6.694 2.17e-11 ***
## age           0.022851   0.009894   2.310  0.02091 *
## dis           0.634380   0.207634   3.055  0.00225 **
## rad           0.718773   0.142066   5.059 4.21e-07 ***
## tax          -0.007676   0.002503  -3.066  0.00217 **
## ptratio       0.303502   0.109255   2.778  0.00547 **
## black        -0.012866   0.006334  -2.031  0.04224 *
## medv          0.112882   0.034362   3.285  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 216.22  on 496  degrees of freedom
## AIC: 236.22
##
## Number of Fisher Scoring iterations: 9
```

```r
model.2 <- glm(crimclass ~ . - indus - chas - rm - lstat, data = boston, family = binomial,
               subset = train)
summary(model.2)
```

```
##
## Call:
## glm(formula = crimclass ~ . - indus - chas - rm - lstat, family = binomial,
##     data = boston, subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3146  -0.3041  -0.0390   0.1653   3.3830
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.078705   6.924853  -5.354 8.58e-08 ***
## zn           -0.069092   0.032505  -2.126  0.03354 *
## nox          48.551533   7.461723   6.507 7.68e-11 ***
## age           0.011921   0.010410   1.145  0.25215
## dis           0.571056   0.211047   2.706  0.00681 **
## rad           0.651362   0.163957   3.973 7.10e-05 ***
## tax          -0.002903   0.003510  -0.827  0.40817
## ptratio       0.377143   0.121314   3.109  0.00188 **
## black        -0.009162   0.006100  -1.502  0.13314
## medv          0.114768   0.036731   3.125  0.00178 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.30  on 378  degrees of freedom
## Residual deviance: 190.08  on 369  degrees of freedom
## AIC: 210.08
```

```
##
## Number of Fisher Scoring iterations: 8
```

```
prob.2 <- predict(model.2, boston.test, type = "response")
pred.2 <- rep(0, length(prob.2))
pred.2[prob.2 > 0.5] = 1
mean(pred.2 != crim.test)
```

```
## [1] 0.07874016
```

Create a new column called “crimclass”. If crime rate is above the median, crimclass is 1, and 0 otherwise. 75% of dataset is training data and 25% is test dataset. The error rate when using all predictors is 7.874016%. Using “bestglm” package we find the best logistic regression model selected from all predictors. With 9 of the predictors, the new logistic regression model also have 7.874016% error rate.

MNIST

Problem 1

```
load("mnist_all.RData")
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.4.3

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

##
## Attaching package: 'pROC'

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

```
index <- (train$y == 0 | train$y == 1)
df <- train$x[index,]
df.y <- train$y[index]
df <- as.data.frame(df)
df$y <- df.y
var(df[,269])
```

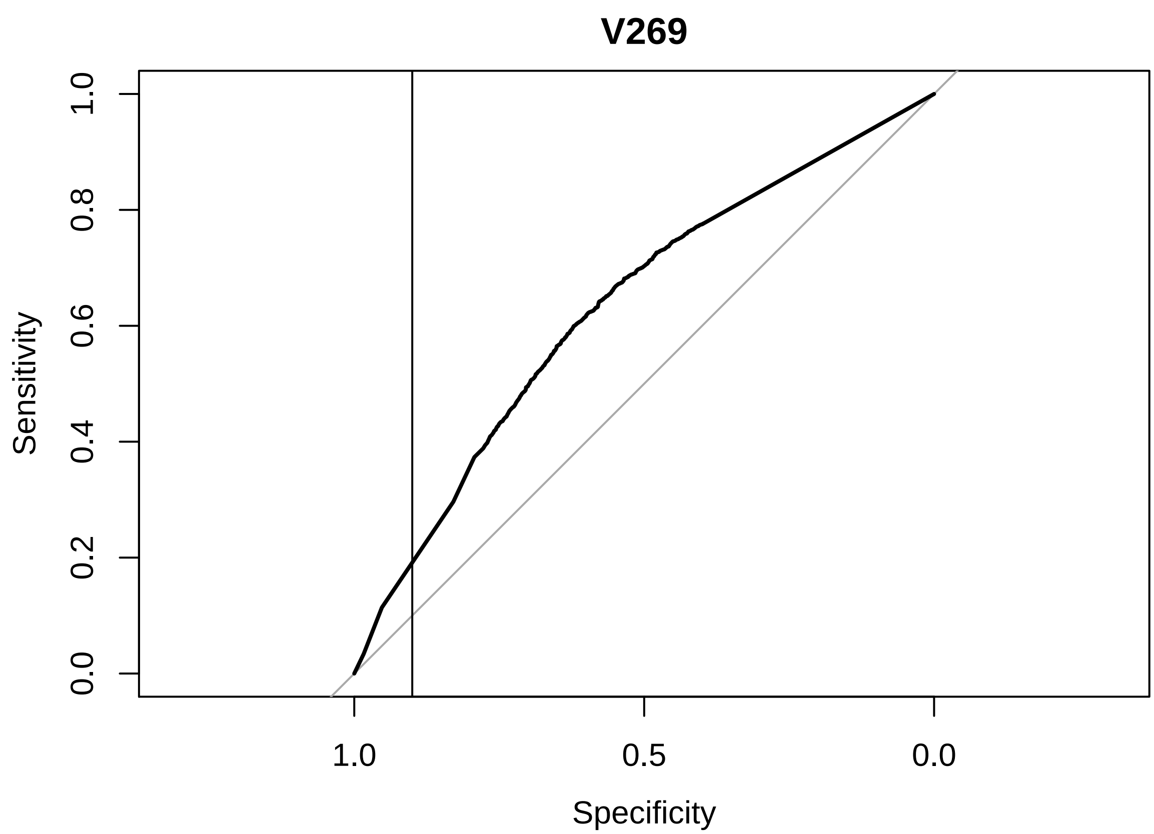
```
## [1] 12159.13
```

```
model.3 <- glm(y ~ V269, data = df, family = binomial)
summary(model.3)
```

```
##
## Call:
## glm(formula = y ~ V269, family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4761  -1.0599   0.9072   1.0202   1.3606
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.420932   0.027898  -15.09   <2e-16 ***
```

```
## V269          0.004315   0.000167   25.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 17504  on 12664  degrees of freedom
## Residual deviance: 16814  on 12663  degrees of freedom
## AIC: 16818
##
## Number of Fisher Scoring iterations: 4
```

```r
df$pred <- predict(model.3, type = "response")
myroc <- roc(df$y, df$pred)
plot(myroc, main = "V269")
abline(v = 0.9)
```




```r
p = .6616
mytable = table(df$y , df$pred > p)
x = c(mytable[2,2]/sum(mytable[2,]), mytable[1,2]/sum(mytable[1,]))
names(x) <- c("trueP", "falseP")
x
```

```
##     trueP    falseP
## 0.2963512 0.1708594
```

```r
p = .6617
mytable = table(df$y , df$pred > p)
```

```r
x = c(mytable[2,2]/sum(mytable[2,]), mytable[1,2]/sum(mytable[1,]))
names(x) <- c("trueP", "falseP")
x
```

```
##      trueP     falseP
## 0.11391279 0.04761101
```

Variable is Pixel No. 269

Logistic regression equation:

$$\mathrm{P}\{\mathrm{y}\} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 V_{269}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 V_{269}}} = \frac{e^{-0.420932+0.004315 V_{269}}}{1+e^{-0.420932+0.004315 V_{269}}}$$

When the fraction of false positives is 0.1,

$$\frac{0.1-0.04761101}{0.1708594-0.04761101} \times (0.2963512-0.11391279)+0.11391279 = 0.1914616$$

the fraction of true positives is approximately 0.19

Problem 2

```r
var(df[,431])
```

```
## [1] 7030.701
```

```r
var(df[,547])
```

```
## [1] 12423.97
```

```r
cor(df[,431], df[,547])
```

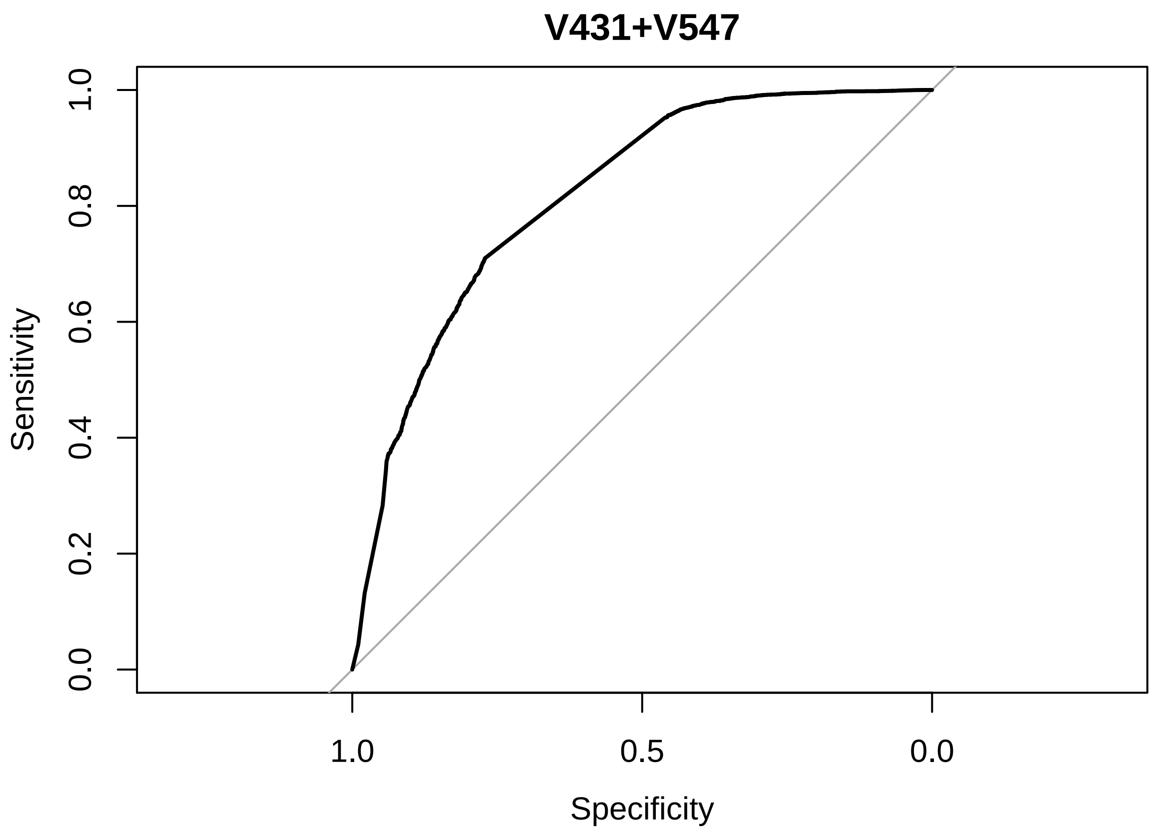
```
## [1] -0.02477022
```

```r
model.4 <- glm(y ~ V431 + V547, data = df, family = binomial)
summary(model.4)
```

```
##
## Call:
## glm(formula = y ~ V431 + V547, family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8892  -1.2011   0.6079   0.8417   3.3224
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.0555008  0.0284174   1.953   0.0508 .
## V431        -0.0219321  0.0006467 -33.914   <2e-16 ***
## V547         0.0060600  0.0002017  30.045   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 17504  on 12664  degrees of freedom
## Residual deviance: 12956  on 12662  degrees of freedom
## AIC: 12962
##
## Number of Fisher Scoring iterations: 6
```

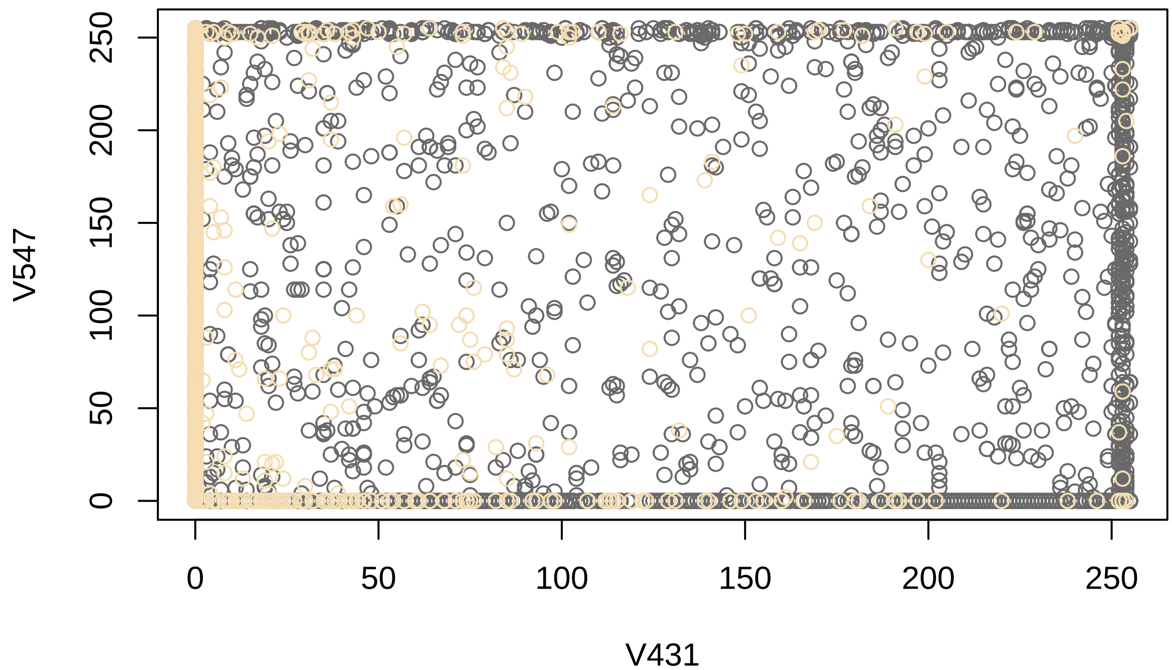
```
df$pred <- predict(model.4, type = "response")
myroc <- roc(df$y, df$pred)
plot(myroc, main = "V431+V547")
```




```
auc(df$y, df$pred)
```

```
## Area under the curve: 0.8166
```

```
plot(df$V431[df$y == 0], df$V547[df$y == 0], col = "dimgrey", xlab = "V431", ylab = "V547")
points(df$V431[df$y == 1], df$V547[df$y == 1], col = "wheat")
```



Classifier ssing Pixel No.431 and No.547 is good. From the scatterplot, major points at bottomleft are 1s and those at upperright are 0s. The training accuracy is 81.66%.

Problem 3

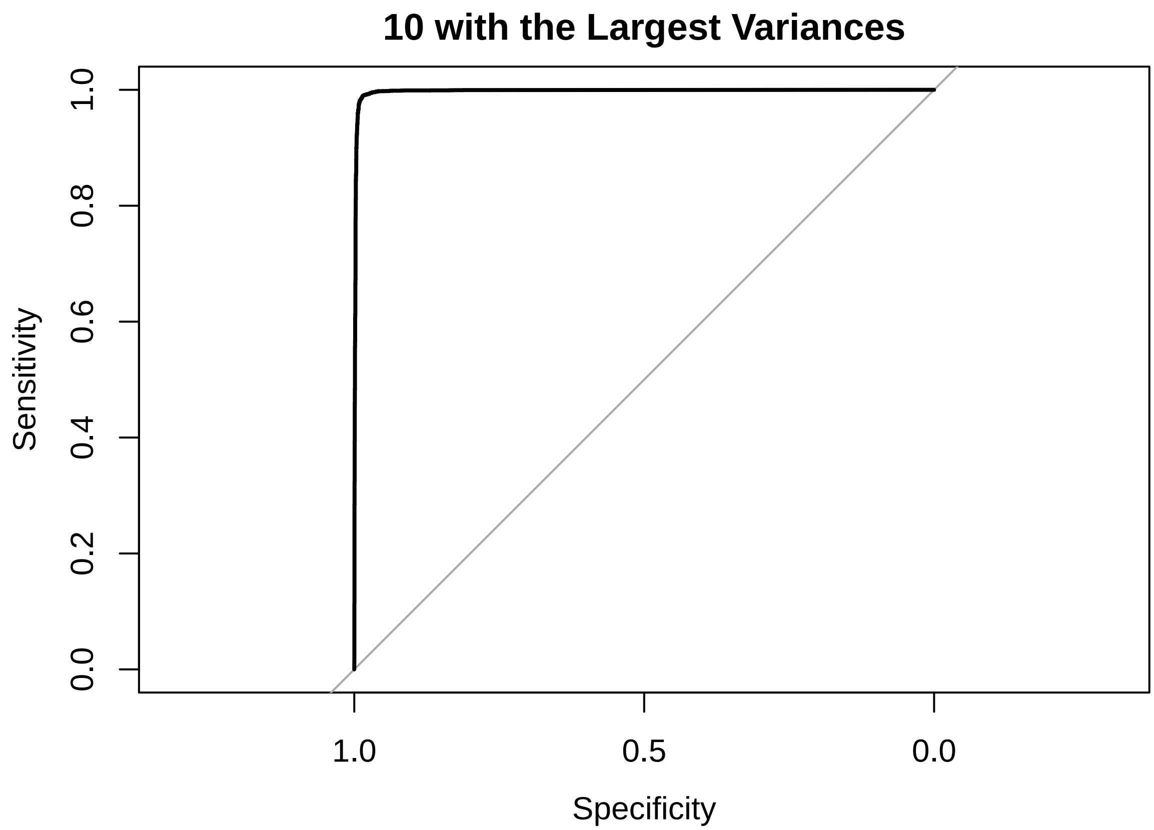
```
variance <- c()
index <- 1:784
for (i in index){
  variance <- c(variance, var(df[,i]))
}
df.var <- data.frame(index, variance)
df.var <- df.var[order(-variance),]
head(df.var, 10)
```

```
##     index variance
## 407   407 15407.78
## 435   435 15163.62
## 379   379 14902.05
## 463   463 14224.23
## 462   462 14008.45
## 352   352 13953.82
## 351   351 13670.39
## 380   380 13664.89
## 490   490 13651.83
## 434   434 13541.13
```

```r
model.5 <- glm(y ~ V351 + V352 + V379 + V380 + V407 + V434 + V435 + V462 + V463 + V490, data = df,
               family = binomial)
summary(model.5)
```

```
##
## Call:
## glm(formula = y ~ V351 + V352 + V379 + V380 + V407 + V434 + V435 +
##     V462 + V463 + V490, family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9196  -0.0958   0.0320   0.0535   3.2821
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.381427   0.173351 -31.043  < 2e-16 ***
## V351         0.002624   0.001865   1.407 0.159352
## V352         0.001963   0.001781   1.102 0.270501
## V379         0.002766   0.002561   1.080 0.280096
## V380         0.008275   0.001852   4.468 7.90e-06 ***
## V407         0.005319   0.002554   2.083 0.037295 *
## V434         0.007240   0.001893   3.823 0.000132 ***
## V435         0.006364   0.002467   2.579 0.009904 **
## V462         0.006390   0.002615   2.444 0.014531 *
## V463         0.011875   0.001722   6.898 5.27e-12 ***
## V490        -0.001590   0.001871  -0.850 0.395532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 17504.4  on 12664  degrees of freedom
## Residual deviance:  1130.5  on 12654  degrees of freedom
## AIC: 1152.5
##
## Number of Fisher Scoring iterations: 8
```

```r
df$pred <- predict(model.5, type = "response")
myroc <- roc(df$y, df$pred)
plot(myroc, main = "10 with the Largest Variances")
```




```r
auc(myroc)
```

```
## Area under the curve: 0.9977
```

The new ROC curve performs better than previous one. The training accuracy is 99.77%.