MAST90138 Assignment 3

867492 Haonan Zhong

Question 1a

```
library(MASS)
suppressMessages(library(pls))
XGtrain <- read.table("XGtrainRain.txt", header = TRUE, sep = ",")
XGtest <- read.table("XGtestRain.txt", header = TRUE, sep = ",")
```

```
# Fitting the quadratic discriminant model
# qda_model <- qda(G~., data = XGtrain)
# Fitting the logistic regression model
logistic <- glm(G~., data = XGtrain, family = binomial(link = "logit"))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
# summary(logistic)
```

When attempting to fit the quadratic discriminant model with all the p predictors in the training set, an error was shown that some groups are too small for fitting, and from the summary of the logistic regression fitted with all p predictors, the model is overfitted as it has zero degrees of freedom. And only 149 of the 365 explanatory variables were used to fit the model, mainly because we only have 150 instances in the training set, thus there are insufficient degrees of freedom to fit all p predictors. Therefore, it is not recommended to use these two classifiers on the test set.

Question 1b

```
# Obtain the projection matrix for PCA
Xtrain <- scale(XGtrain[, -c(366)], scale=FALSE)
PCX <- prcomp(Xtrain, retx = T)
gamma <- PCX$rotation
# Manually re-compute the PC components
Y <- (Xtrain - matrix(rep(1, nrow(Xtrain)), nrow=nrow(Xtrain)) %*% colMeans(Xtrain)) %*% gamma
all(PCX$x == Y)
```

```
## [1] TRUE
```

```
# Obtain the projection matrix for PLS
PLS <- plsr(G~., data = XGtrain)
phi <- PLS$projection
t <- Xtrain %*% phi
all(PLS$scores == t)
```

```
## [1] TRUE
```

As we can see from the output above, all the manually computed components are the same as the one outputed by the R function.

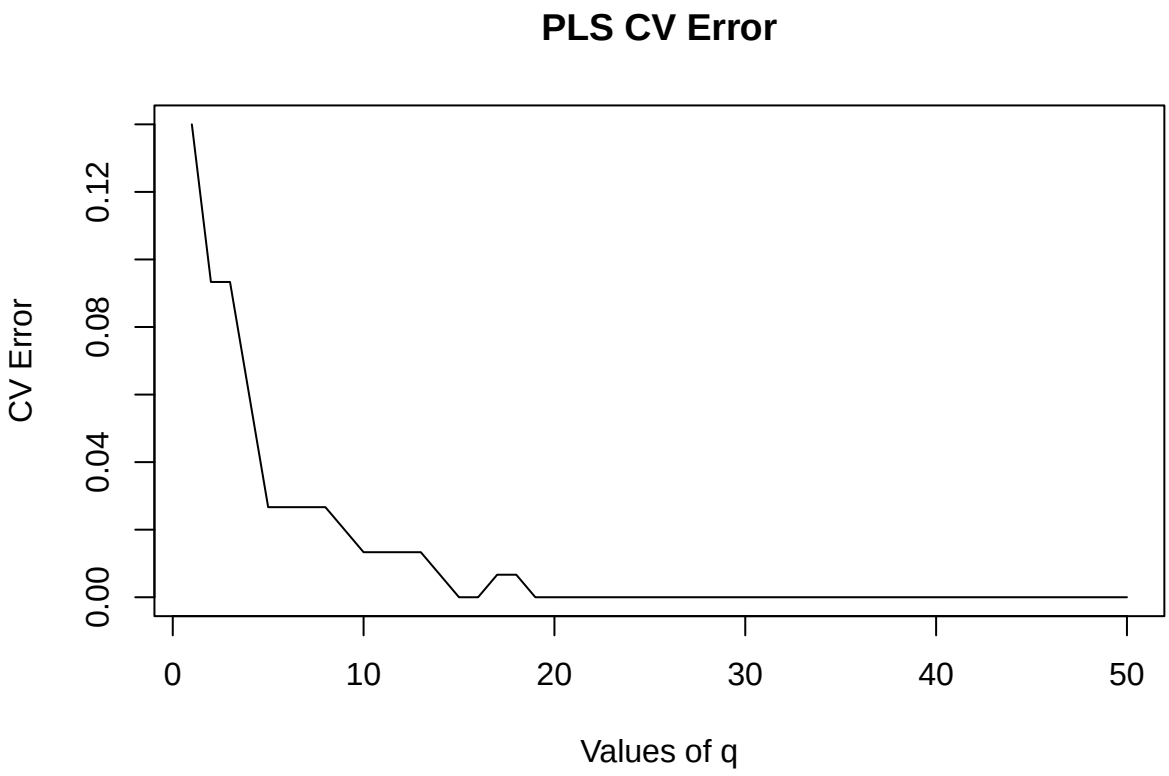
Question 1c

```r
# Train QDA with PLS components
CV_error <- rep(0, 50)
Gtrain <- XGtrain[, 366]

for (q in 1:50) {
  prediction <- c()
  for (i in 1:dim(XGtrain)[1]) {
    GDATACV <- Gtrain[-i]
    YDATACV <- as.data.frame(PLS$scores[-i, 1:q])
    QDA <- qda(GDATACV~., data = YDATACV)

    new_data <- as.data.frame(t(PLS$scores[i, 1:q]))
    colnames(new_data) <- colnames(YDATACV)
    prediction[i] <- as.numeric(predict(QDA, newdata = new_data)$class) - 1
  }
  CV_error[q] <- sum(prediction != Gtrain)/dim(XGtrain)[1]
}

# Plotting the cross validation error
plot(c(1:50), CV_error, type = "l", xlab = "Values of q", ylab = "CV Error", main = "PLS CV Error")
```




```
which(CV_error == min(CV_error))
```

```
##  [1] 15 16 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
## [26] 42 43 44 45 46 47 48 49 50
```

```
min(CV_error)
```

```
## [1] 0
```

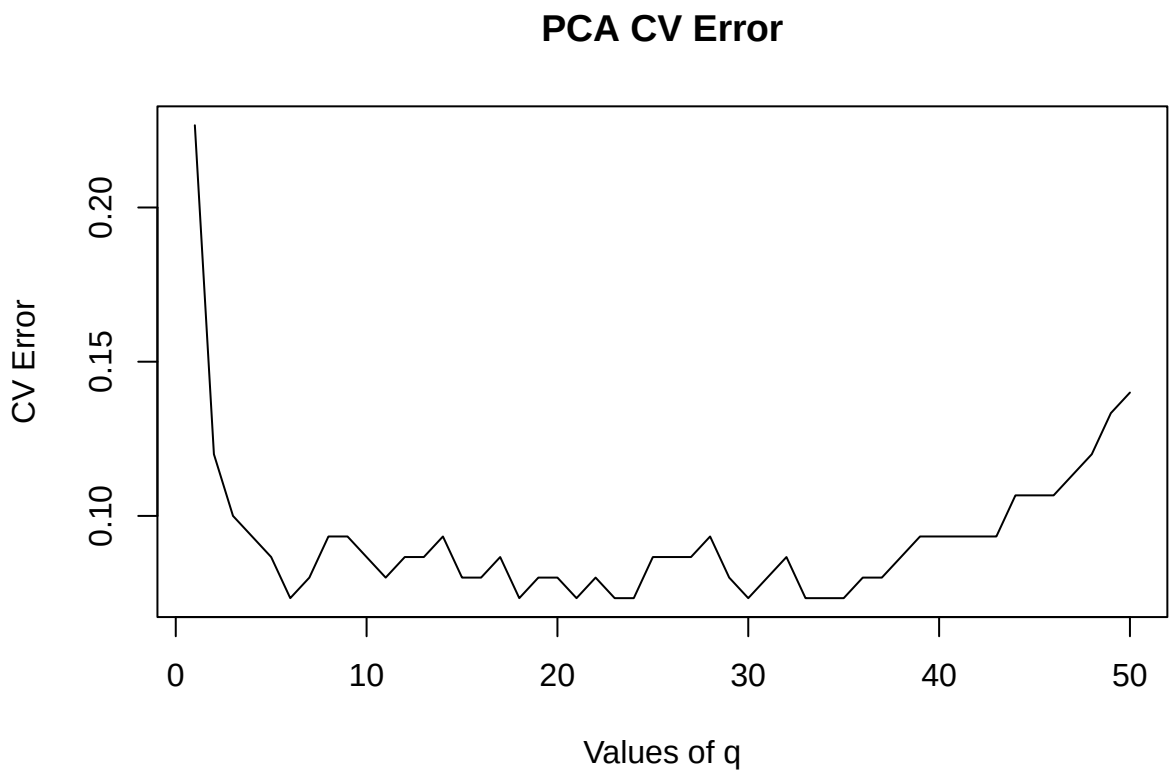
As we can see from the result above, cross validation error are equal to zero and lowest when $q = 15$. Although there are numerous values for q that leads to the lowest error, but in order to keep the model simple, thus, the chosen number of components for PLS should be $q = 15$.

```
# Train QDA with PCA components
CV_error <- rep(0, 50)
for (q in 1:50) {
  prediction <- c()
  for (i in 1:dim(XGtrain)[1]) {
    GDATACV <- Gtrain[-i]
    YDATACV <- as.data.frame(PCX$x[-i, 1:q])
    QDA <- qda(GDATACV~., data = YDATACV)

    new_data <- as.data.frame(t(PCX$x[i, 1:q]))
    colnames(new_data) <- colnames(YDATACV)
```

```
    prediction[i] <- as.numeric(predict(QDA, newdata = new_data)$class) - 1
  }
  CV_error[q] <- sum(prediction != Gtrain)/dim(XGtrain)[1]
}

# Plotting the cross validation error
plot(c(1:50), CV_error, type = "l", xlab = "Values of q", ylab = "CV Error", main = "PCA CV Error")
```




```
which(CV_error == min(CV_error))
```

```
## [1]  6 18 21 23 24 30 33 34 35
```

```
min(CV_error)
```

```
## [1] 0.07333333
```

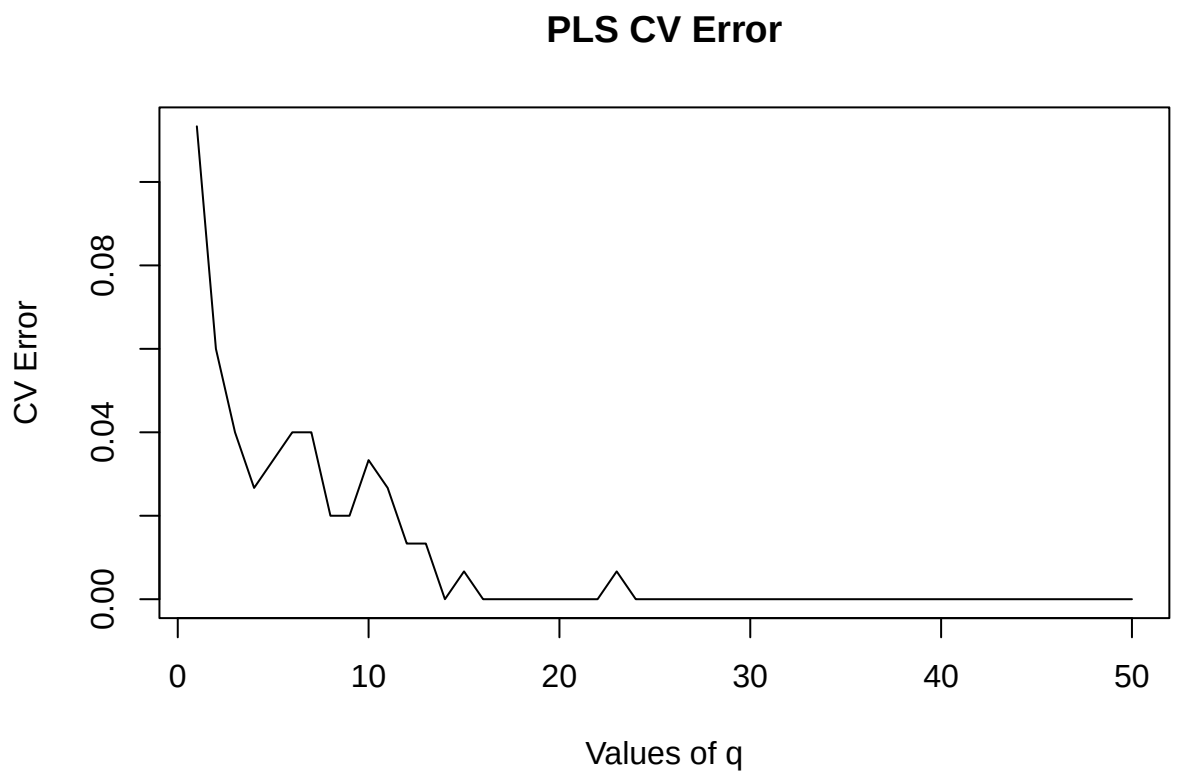
As we can see from the output above, cross validation error are equal to 0.073 and lowest when $q = 6$. Although there are numerous values for q that leads to the lowest error, but in order to keep the model simple, thus, the chosen number of components for PCA should be $q = 6$.

```
# Train logistic regression model with PLS
CV_error <- rep(0, 50)
for (q in 1:50) {
```

```
  prediction <- c()
  for (i in 1:dim(XGtrain)[1]) {
    GDATACV <- Gtrain[-i]
    YDATACV <- as.data.frame(PLS$scores[-i, 1:q])
    suppressWarnings(logistic <- glm(GDATACV~., data = YDATACV, family = binomial(link = "logit")))

    new_data <- as.data.frame(t(PLS$scores[i, 1:q]))
    colnames(new_data) <- colnames(YDATACV)
    prediction[i] <- ifelse(predict(logistic, newdata = new_data) > 0, 1, 0)
  }
  CV_error[q] <- sum(prediction != Gtrain)/dim(XGtrain)[1]
}

# Plotting the cross validation error
plot(c(1:50), CV_error, type = "l", xlab = "Values of q", ylab = "CV Error", main = "PLS CV Error")
```




```
which(CV_error == min(CV_error))
```

```
##  [1] 14 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## [26] 41 42 43 44 45 46 47 48 49 50
```

```
min(CV_error)
```

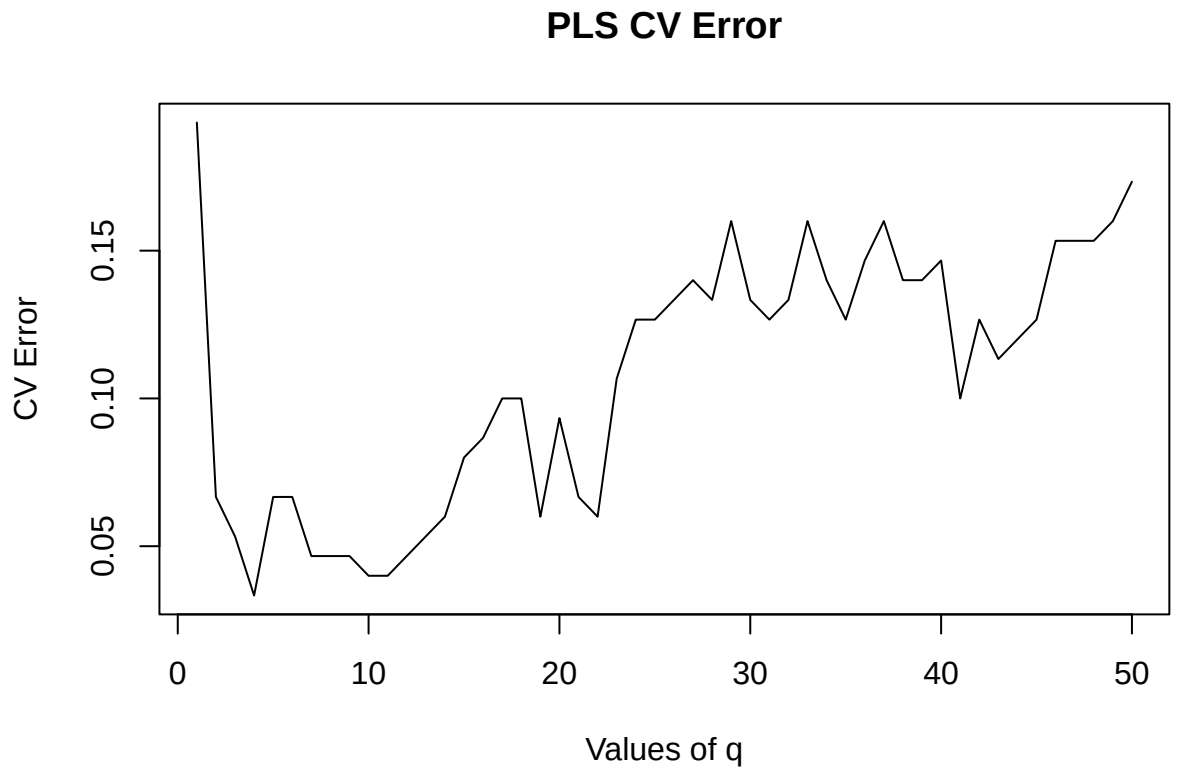
```
## [1] 0
```

As we can see from the result above, cross validation error are equal to zero and lowest when $q = 14$. Although there are numerous values for q that leads to the lowest error, but in order to keep the model simple, thus, the chosen number of components for PLS should be $q = 14$.

```
# Train logistic regression model with PCA components
CV_error <- rep(0, 50)
for (q in 1:50) {
  prediction <- c()
  for (i in 1:dim(XGtrain)[1]) {
    GDATACV <- Gtrain[-i]
    YDATACV <- as.data.frame(PCX$x[-i, 1:q])
    suppressWarnings(logistic <- glm(GDATACV~., data = YDATACV, family = binomial(link = "logit")))

    new_data <- as.data.frame(t(PCX$x[i, 1:q]))
    colnames(new_data) <- colnames(YDATACV)
    prediction[i] <- ifelse(predict(logistic, newdata = new_data) > 0, 1, 0)
  }
  CV_error[q] <- sum(prediction != Gtrain)/dim(XGtrain)[1]
}

# Plotting the cross validation error
plot(c(1:50), CV_error, type = "l", xlab = "Values of q", ylab = "CV Error", main = "PLS CV Error")
```




```
which(CV_error == min(CV_error))
```

```
## [1] 4
```

```
min(CV_error)
```

```
## [1] 0.03333333
```

As we can see from the result above, cross validation error are equal to 0.033 and lowest when $q = 4$. Thus, the chosen number of components for PLS should be $q = 4$.

Question 1d

Based on the results of question 1c, we can see that for both quadratic discriminant model and logistic regression model, with the optimal q, PLS tend to give the lowest cross validation error equals to 0, which is better than each of the classifiers with PCA. Therefore, we would prefer PLS version of the classifiers.

Question 1e

```
# Retrain all the classifiers with the suggested value of q and the full training set
QDA_PLS <- qda(Gtrain~., data = as.data.frame(PLS$scores[, 1:15]))
QDA_PCA <- qda(Gtrain~., data = as.data.frame(PCX$x[, 1:6]))

logistic_PLS <- glm(Gtrain~., data = as.data.frame(PLS$scores[, 1:14]), family = binomial(link = "logit
logistic_PCA <- glm(Gtrain~., data = as.data.frame(PCX$x[, 1:4]), family = binomial(link = "logit"))
```

```
# Prepare the test set
Xtest <- XGtest[,-c(366)]
Gtest <- XGtest[, 366]
repbarX <- matrix(rep(colMeans(XGtrain), dim(Xtest)[1]), nrow = dim(Xtest)[1], byrow = T)
Y_new <- as.matrix(Xtest - repbarX) %*% gamma

# Report test error for quadratic discriminant with PCA components
PCA_newdata <- as.data.frame(Y_new[, 1:6])
prediction <- predict(QDA_PCA, newdata = PCA_newdata)$class
QDA_PCA_ERROR <- sum(prediction != Gtest)/dim(Xtest)[1]
paste("Test error for quadratic discriminant with PCA components is", QDA_PCA_ERROR)
```

```
## [1] "Test error for quadratic discriminant with PCA components is 0.0975609756097561"
```

```
# Report test error for logistic regression with PCA components
PCA_newdata <- as.data.frame(Y_new[, 1:4])
prediction <- ifelse(predict(logistic_PCA, newdata = PCA_newdata) > 0, 1, 0)
LR_PCA_ERROR <- sum(prediction != Gtest)/dim(Xtest)[1]
paste("Test error for logistic regression with PCA components is", LR_PCA_ERROR)
```

```
## [1] "Test error for logistic regression with PCA components is 0.024390243902439"
```

```
# Report test error for quadratic discriminant with PLS components
t_new <- as.matrix(Xtest - repbarX) %*% phi
PLS_newdata <- as.data.frame(t_new[, 1:15])
prediction <- predict(QDA_PLS, newdata = PLS_newdata)$class
QDA_PLS_ERROR <- sum(prediction != Gtest)/dim(Xtest)[1]
paste("Test error for quadratic discriminant with PLS components is", QDA_PLS_ERROR)
```

```
## [1] "Test error for quadratic discriminant with PLS components is 0.0975609756097561"
```

```r
# Report test error for logistic regression with PLS components
PLS_newdata <- as.data.frame(t_new[, 1:14])
prediction <- ifelse(predict(logistic_PLS, newdata = PLS_newdata) > 0, 1, 0)
LR_PLS_ERROR <- sum(prediction != Gtest)/dim(Xtest)[1]
paste("Test error for logistic regression with PLS components is", LR_PLS_ERROR)
```

```
## [1] "Test error for logistic regression with PLS components is 0.121951219512195"
```

As we can see from the test errors for the four classifiers, quadratic discriminant produces similar test error with both PCA and PLS component. Whilst logistic regression with PCA component produces a test error that's much smaller than logistic regression with PLS. And it is worth mentioning that for both quadratic discriminant and logistic regression, they tend to use less number of PCA components, while producing similar or better test error than PLS components.