Homework Assignment 2
Varada Gurjar 1113815
Naga Padmavathy Kancharla

1.

a)

```
Set the working directory to the path for az-5000.txt
az <- read.table("az-5000.txt",header=TRUE)
head<-head(az,n=1)
tail<-tail(az,n=1)                                  #The head and tail look as shown in the question.
```

b)

```
subpart <- sample(1:nrow(az),size=0.8*nrow(az))
training <- az[subpart,]                            # training data created from 80% of the data.
dim(training)                                       #we see training data has 4000 rows.

test<-az[-subpart,]                                 # test data is complement of the training data.
```

c)

```
summary <- table(training[1])                       # summary contains cases per class of training data
```

2.

a) We replicate the 1/26 value for each class to create a prior vector

```
prior <- rep(c(1/26),each=26)
```

b)

```
install.packages("MASS")
library(MASS)
azlda <- lda(char ~.,training,prior=prior1)                                  # calculate the lda
```

c)

```
predict <- predict(azlda, test)$class
confusionMat <- table(test$char, predict(azlda, test)$class)                 # create confusion matrix
confusionMat <- table(test$char,predict)
dim(confusionMat)

colSums(confusionMat)-diag(confusionMat)/colSums(confusionMat)
```

d)

```
length(which(predict(azlda, test)$class == test$char))                       #accuracy of the test data

length(which(predict(azlda, training)$class == training$char))               #accuracy of the training data
```

3.

```
Set the working directory to the path for credit_data.txt
credit <- read.table("credit_data.txt",header=TRUE)
dim(credit)                                         #We see 885 observations or bank firm information

subcredit <- sample(1:nrow(credit),size=0.8*nrow(credit))
```

```
credit_training <- credit[subcredit,]
credit_test<-credit[-subcredit,]

table(credit_training$Fail)                #showing the number of cases per class for both training and test data

table(credit_test$Fail)

a)
glmcredit <- glm(Fail ~., data = credit_training, family = binomial)
summary(glmcredit)

c)
testPredict <- predict(glmcredit , credit_test, type = "response")
table(credit_test$Fail, testPredict >= 0.5)
```

4.

```
Set the working directory to the path for credit_data.txt
credit <- read.table("credit_data.txt",header=TRUE)
dim(credit)

subcredit <- sample(1:nrow(credit),size=0.8*nrow(credit))
credit_training <- credit[subcredit,]
credit_test<-credit[-subcredit,]

x <- as.matrix(credit[subcredit , 3:15])                  # safe casting 80% credit data frame values to a matrix
y <- 2*credit$Fail[subcredit ]-1                                  #mapping y from the 0-1 to -1 to 1 range

a)
                                                        # fitting regularized logistic regression to training data
credit.glmnet <- cv.glmnet(x, y, family = "binomial")
plot(credit.glmnet)                                               #plotting the cross validation curve

b)
coefVector <- coef(credit.glmnet, lambda = credit.glmnet$lambda.1se)
print(coefVector)

c)
x.test <- as.matrix(credit[-subcredit , c(3:15)])
y.test <- credit_training$Fail

testPredict <- as.numeric(predict(credit.glmnet, x.test, type = "class", lambda = credit.glmnet$lambda.1se))

confusionMat <- table(y.test, testPredict )                       #creating confusion matrix

accuracy <- sum(diag(confusionMat ))/sum(confusionMat )                  #calculating accuracy
```

5.

a)

If x lies in the interval [0.05,0.95] then observations we use are in [x−0.05,x+0.05] interval representing a length of 0.1 and a fraction of 10%.

For values of x < 0.05, we use observations in [0,x+0.05] interval. This represents a fraction of (100x+5)%; Similarly if x>0.95, then observations we use is (105−100x)%.

To compute the average fraction we will use to make the prediction we have to take area under the curve between 0 to 0.05, 0.05 to 0.95 and 0.95 to 1 as below.

$$\int_0^{0.05} (100x+5)dx + \int_{0.05}^{0.95} 10dx + \int_{0.95}^{1} (105-100x)dx = 0.375 + 9 + 0.375 = 9.75$$

On an average, the fraction of available observations we use to make the prediction is 9.75%.

b)

We can assume observations X1 and X2 to be independent, the fraction of available observations we will use to make the prediction is 9.75%×9.75% = 0.950625%.

c)

With the same argument as before, we may conclude that the fraction of available observations we will use to make the prediction is 9.75^{100} % ≃ 0%

d)

As we saw before, the fraction of available observations we will use to make the prediction is $(9.75)^p$ % with p as the number of features. So when $p \to \infty$, we have

$$\lim_{p \to \infty} (9.75)^p \% = 0.$$

e) For p=1, we have l =0.1,

For p=2, we have $l=0.1^{1/2}$ and for p=100, we have $l=0.1^{1/100}$.

---

6)

a) set.seed(1)

```
y = rnorm(100)
x = rnorm(100)
y = x - 2 * x^2 + rnorm(100)
```

n = 100, p = 2.

b) plot(x, y)

c) library(boot)

```
Data = data.frame(x, y)
set.seed(1)

# i.
glm.fit = glm(y ~ x)
cv.glm(Data, glm.fit)$delta
```

```
# ii.
glm.fit = glm(y ~ poly(x, 2))
cv.glm(Data, glm.fit)$delta

# iii.
glm.fit = glm(y ~ poly(x, 3))
cv.glm(Data, glm.fit)$delta

# iv.
glm.fit = glm(y ~ poly(x, 4))
cv.glm(Data, glm.fit)$delta
```

d)

```
set.seed(15)
# i.
glm.fit = glm(y ~ x)
cv.glm(Data, glm.fit)$delta

# ii.
glm.fit = glm(y ~ poly(x, 2))
cv.glm(Data, glm.fit)$delta

# iii.
glm.fit = glm(y ~ poly(x, 3))
cv.glm(Data, glm.fit)$delta

# iv.
glm.fit = glm(y ~ poly(x, 4))
cv.glm(Data, glm.fit)$delta
```

The result is the same, because LOOCV will be the same since it evaluates n folds of a single observation.

e)

The quadratic polynomial had the lowest LOOCV test error rate. This was expected because it matches the true form of Y.

f)

```
summary(glm.fit)
```
