Problem 1 (Ex2.2 pg 39)

Solution:
The Bayes classifier is \( \hat{G}(X) = \max_{g \in G} \Pr(g|X = x) \). To find the optimal boundary, we just need to find where
\[ \Pr(g = 0|X = x) = \Pr(g = 1|X = x) = 1/2, \]
which is \( \Leftrightarrow \)
\[ \Pr(X = x|g = 0) \Pr(g = 0) = \Pr(X = x|g = 1) \Pr(g = 1) \]
Since the generating density \( \Pr(X = x|g) \) is known, prior \( \Pr(g) \) is based on our prior knowledge, for example, we can set \( \Pr(g = 1) = \Pr(g = 0) = 1/2 \), we can calculate the boundary exactly.

Problem 2 (Ex2.3 pg 39)

Solution:
Define \( r \) as the median distance from the origin to the closest data point. Then
\[ \Pr(\text{all N points have distance} \geq r) = 1/2. \]
For each point \( x_i \) in the unit ball, denote the distance to the origin \( d_i \),
\[ \Pr(d_i \geq r) = 1 - \Pr(d_i < r) = 1 - \frac{c\pi r^p}{c\pi 1^p} = 1 - r^p \]
Therefore, we have
\[ \Pr(\text{all N points have distance} \geq r) = \prod_{i=1}^N \Pr(d_i \geq r) = (1 - r^p)^N = 1/2 \]
Solve the equation, we get \( r = (1 - \frac{1/2}{N})^{1/p} \)

Problem 3 (Ex2.4 pg 39)

Solution:
(a). \( a = \frac{x_0}{||x_0||}, \quad z_i = a^T x_i, \)
\[ \mathbb{E}(z_i) = \mathbb{E}[a^T x_i] = a^T \mathbb{E}[x_i] = a^T \cdot 0 = 0 \]
\[ \text{Var}(z_i) = \text{Var}[a^T x_i] = a^2 \text{Var}[x_i] = a^2 \cdot 1 = \sum_{k=1}^p a_k^2 = 1 \]
Linear combination of normals is still normal, \( z_i \sim \text{No}(0,1) \), thus the squared distance from origin is 1. While for the target point, \( x_i \sim \text{No}(0, I_p) \), the squared distance has a \( \chi^2 \) distribution with mean \( p \).

(b). Apply results from (a), the expected square distance of a test point from the expected center point of the training data, which is origin, is \( p \). The expected distance is \( \sqrt{p} \), while along direction \( a \), all training points have expected distance 1.

Problem 4 (Ex2.8 pg 40)

Solution:

<table>
<thead>
<tr>
<th>Table 1: Classification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression training error:</td>
</tr>
<tr>
<td>Linear Regression test error:</td>
</tr>
<tr>
<td>1-nearest neighbour training error:</td>
</tr>
<tr>
<td>1-nearest neighbour test error:</td>
</tr>
<tr>
<td>15-nearest neighbour training error:</td>
</tr>
<tr>
<td>15-nearest neighbour test error:</td>
</tr>
</tbody>
</table>

```r
## Ex 2.7
## Download dataset from the web.
## Read in data
train.2 <- read.table("train.2",sep=";")
train.3 <- read.table("train.3",sep=";")
training <- rbind(train.2,train.3)
y.obs <- c(rep(2,nrow(train.2)),rep(3,nrow(train.3)))

test <- read.table("zip.test")
test.2 <- test[test[,1]==2,]
test.3 <- test[test[,1]==3,]
test <- rbind(test.2,test.3)
y.test <- test[,1];
test <- test[,2:(ncol(test))]
colnames(test) <- colnames(training)

## Least Square Estimation

lm1 <- lm(y.obs~.,data=training)
summary(lm1)
y.hat <- lm1$fitted
summary(y.hat)
## Suppose we use 2.5 as the seperating point,
ind3 <- (y.hat>=2.5)
ind2 <- (y.hat< 2.5)
y.hat[ind3] <- 3;
y.hat[ind2] <- 2;

cat("Linear Regression training error:", mean((y.hat-y.obs)^2),"n")
```

2
y.hat.test <- predict.lm(lm1,test);
## Again we use 2.5 as the seperating point,
ind3 <- (y.hat.test>=2.5)
ind2 <- (y.hat.test< 2.5)
y.hat.test[ind3] <- 3;
y.hat.test[ind2] <- 2;
cat("Linear Regression test error: ", mean((y.hat.test-y.test)^2),"\n")

################
###### 1-nearest neighbour
library(class)
y.hat.1 <- knn(training,training,y.obs,k=1)
cat("1-nearest neighbour training error: ", mean((as.numeric(y.hat.1)-as.numeric(as.factor(y.obs)))^2),"\n")
y.hat.1.test <- knn(training,test,y.obs,k=1)
cat("1-nearest neighbour test error: ", mean((as.numeric(y.hat.1.test)-as.numeric(as.factor(y.test)))^2),"\n")
y.hat.15 <- knn(training,training,y.obs,k=15)
cat("15-nearest neighbour training error: ", mean((as.numeric(y.hat.15)-as.numeric(as.factor(y.obs)))^2),"\n")
y.hat.15.test <- knn(training,test,y.obs,k=15)
cat("15-nearest neighbour test error: ", mean((as.numeric(y.hat.1.test)-as.numeric(as.factor(y.test)))^2),"\n")