Jakob Lovato
Will Bliss
STAT 760
HW 2

1. The boundary is defined where

$$P(\text{Blue} \mid X = x) = P(\text{Orange} \mid X = x).$$

Since $P(\text{Blue} \cap X = x) = P(X = x \mid \text{Blue})P(\text{Blue})$, by Bayes' theorem, we have

$$P(\text{Blue} \mid X = x) = \frac{P(\text{Blue} \cap X = x)}{P(X = x)}$$

$$= \frac{P(X = x \mid \text{Blue})P(\text{Blue})}{P(X = x)}.$$

Similarly,

$$P(\text{Orange} \mid X = x) = \frac{P(X = x \mid \text{Orange})P(\text{Orange})}{P(X = x)}.$$

Then the boundary is defined by

$$\frac{P(X = x \mid \text{Blue})P(\text{Blue})}{P(X = x)} = \frac{P(X = x \mid \text{Orange})P(\text{Orange})}{P(X = x)}$$

$$\Rightarrow P(X = x \mid \text{Blue})P(\text{Blue}) = P(X = x \mid \text{Orange})P(\text{Orange}).$$

Since $P(\text{Blue}) = P(\text{Orange}) = \frac{1}{2}$ at the decision boundary, it follows that the decision boundary can be computed where

$$P(X = x \mid \text{Blue}) = P(X = x \mid \text{Orange}).$$

2. a) Linear regression:

$$\hat{f}(x_0) = x_0^T X(X^T X)^{-1} X^T Y$$
$$= \sum_{i=1}^{n} \left(X(X^T X)^{-1} X^T x_0 \right) y_i$$

Then $l_i(x_0, \mathcal{X}) = (\mathcal{X}(\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T x_0)$.

KNN: $\hat{f}(k_0) = \frac{1}{k} \sum_{i=1}^{n} (f(x_i) + \varepsilon_i) = \frac{1}{k} \sum_{i=1}^{n} y_i$

Then $l_i(x_0, \mathcal{X}) = \frac{1}{k}$ if x_i is within the k-nearest neighbors of the training set

Hence, linear regression and KNN are of this class of estimators.

c) $E_{y,x}(f(x_0) - \hat{f}(x_0))^2 = f(x_0)^2 - 2f(x_0)E_{y,x}(\hat{f}(x_0)) + E_{y,x}(\hat{f}(x_0)^2)$

Since $f(x_0)$ is constant. Then

$$f(x_0)^2 - 2f(x_0)E_{y,x}(\hat{f}(x_0)) + E_{y,x}(\hat{f}(x_0)^2)$$
$$= (f(x_0) - E_{y,x}(\hat{f}(x_0)))^2 + E_{y,x}(\hat{f}(x_0)^2) - (E_{y,x}(\hat{f}(x_0)))^2$$
$$= \text{Bias}(\hat{f}(x_0))^2 + \text{Var}(f(x_0))$$

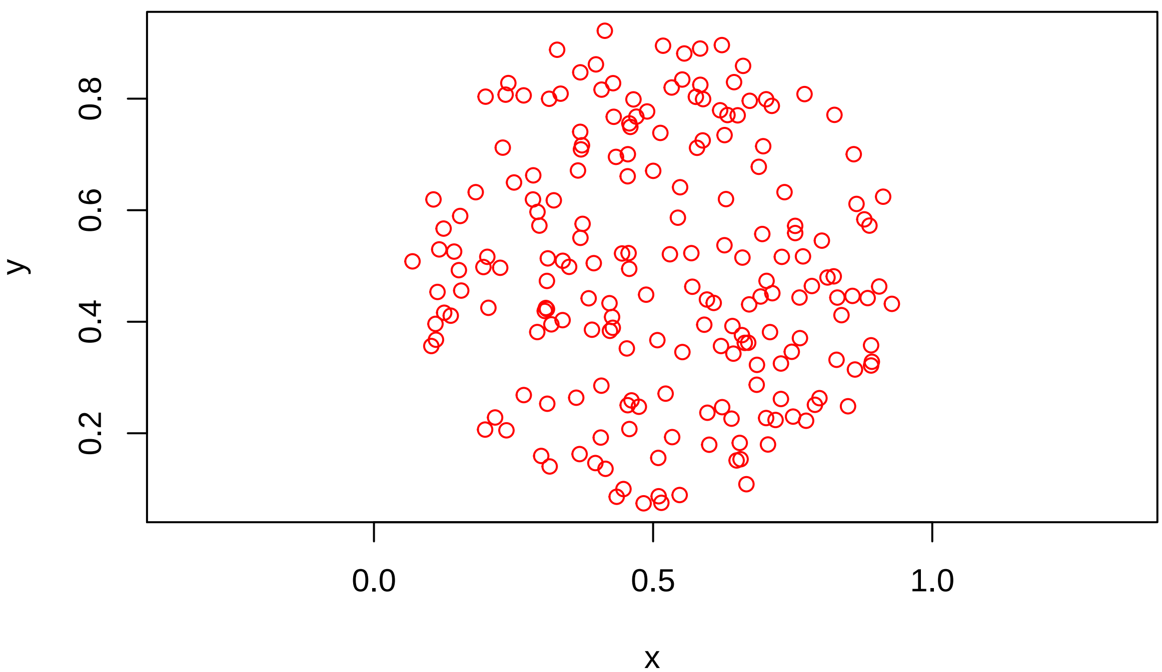
Stat 760 Homework 2 Question 3

Will Bliss and Jakob Lovato

2/8/2022

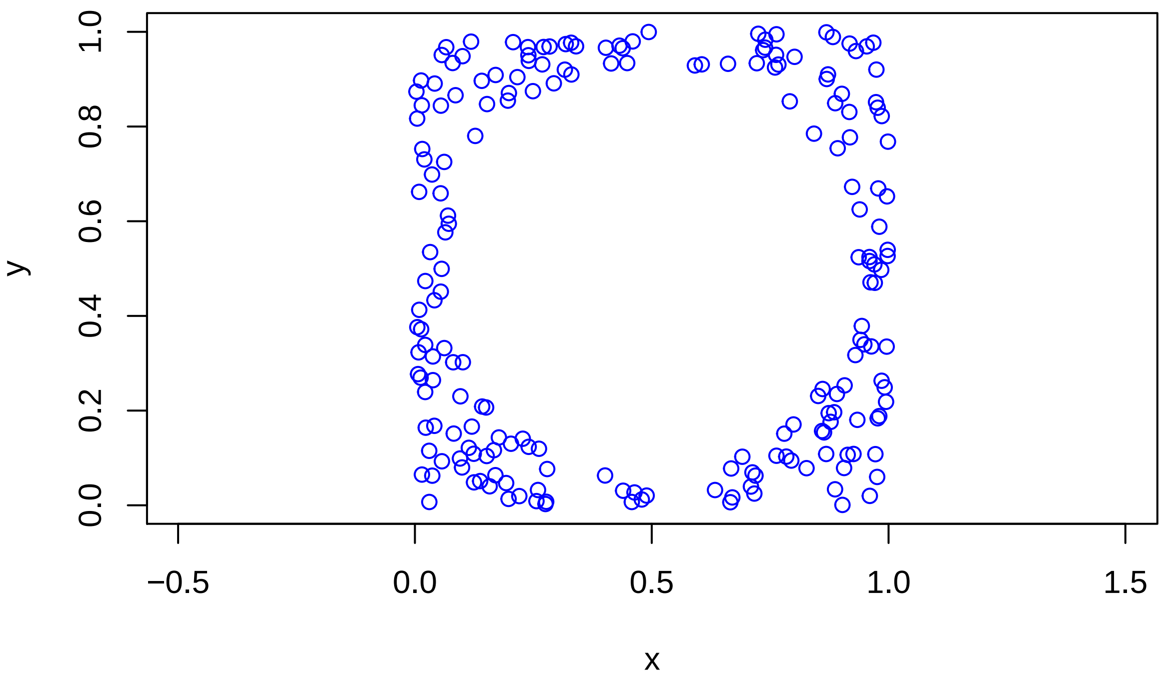
```
###Radius of inner circle
R <- sqrt(0.6 / pi)

###Generate points inside circle
insidex <- c()
insidey <- c()
while(TRUE){
  randx <- runif(1, min = 0, max = 1)
  randy <- runif(1, min = 0, max = 1)
  if((randx - 0.5) ^ 2 + (randy - 0.5) ^2 < (R ^ 2)){
    insidex <- c(insidex, randx)
    insidey <- c(insidey, randy)
  }
  if(length(insidex) >= 200){break}
}
inside <- data.frame(x = insidex, y = insidey)
plot(inside, asp = 1, col = "red")
```




```
###Generate points outside of circle
outsidex <- c()
outsidey <- c()
```

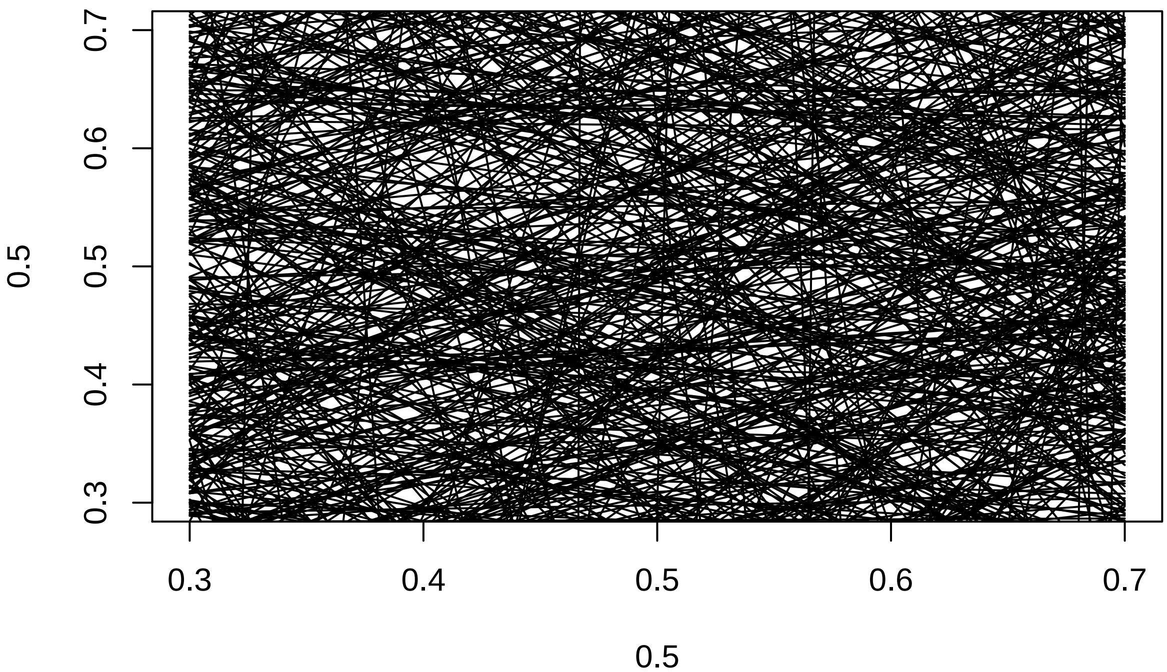
```r
while(TRUE){
  randx <- runif(1, min = 0, max = 1)
  randy <- runif(1, min = 0, max = 1)
  if((randx - 0.5) ^ 2 + (randy - 0.5) ^2 > (R ^ 2)){
    outsidex <- c(outsidex, randx)
    outsidey <- c(outsidey, randy)
  }
  if(length(outsidex) >= 200){break}
}
outside <- data.frame(x = outsidex, y = outsidey)
plot(outside, asp = 1, col = "blue")
```




```r
###Assign true classifications to data
inside <- cbind(inside, class = 1)
outside <- cbind(outside, class = -1)
data <- rbind(inside, outside)
```

```r
###Generate cuts
plot(0.5, 0.5)
hitmiss <- matrix(0, nrow = 400, ncol = 1000)
prediction <- matrix(0, nrow = 400, ncol = 1000)
cuts <- data.frame(a = 1, b = 1, c = 1)
for(i in 1:1000){
  X <- runif(1, min = 0, max = 1)
  y <- runif(1, min = 0, max = 1)
  a <- runif(1, min = -1, max = 1)
  b <- runif(1, min = -1, max = 1)
  c <- a * X + b * y
  ###Pool of all 1,000 cuts (potential committee members)
  curve((c - a * x) / b, add = TRUE)
  cuts[i, 1] <- a
  cuts[i, 2] <- b
  cuts[i, 3] <- c
```

```
  for(j in 1:400){
    ifelse(((a * data[j,]$x) + (b * data[j,]$y)) >= c, prediction[j, i] <- 1, prediction[j, i] <- -1)
  }
}
```



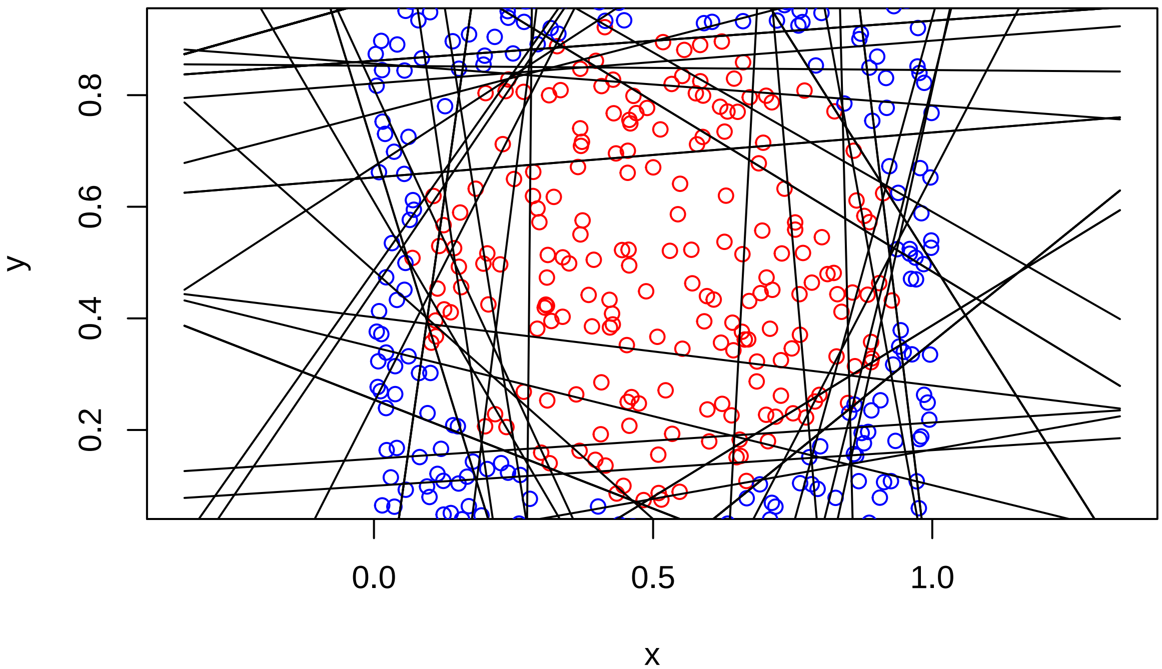

```
for(i in 1:1000){
  for(j in 1:400){
    ifelse(prediction[j, i] == data$class[j], hitmiss[j, i] <- 0, hitmiss[j, i] <- 1)
  }
}
```

```
###ADABoost
weights <- rep(1, 400)
M <- 100
#alpha <- c()
members <- c()
committee <- data.frame(temp = rep(1, 400))
for(m in 1:M){
  W <- sum(weights)
  We <- min(weights %*% hitmiss)
  Wh <- W - We
  alpha <- 0.5 * log(Wh / We)
  member <- which(min(weights %*% hitmiss) == weights %*% hitmiss)
  if(length(member) > 1){
    member <- member[1]
  }
  members <- c(members, member)
  #committee[,m] <- alpha * hitmiss[,member]
  committee <- cbind(committee, alpha * prediction[,member])

  misses <- which(hitmiss[,member] == 1)
  hits <- which(hitmiss[,member] == 0)
  for(i in misses){
```

```r
    weights[i] <- (weights[i] * exp(alpha))
  }
  for(i in hits){
    weights[i] <- (weights[i] * exp(-alpha))
  }
}
committee <- committee[,-1]
```

```r
###Plot selected committee
plot(inside[,-3], asp = 1, col = "red")
points(outside[,-3], asp = 1, col = "blue")
for(i in members){
  curve((cuts$c[i] - cuts$a[i] * x) / cuts$b[i], add = TRUE)
}
```




```r
###Classify the data using ADABoost model
classifications <- sign(rowSums(committee))
error <- mean(data$class != classifications)
error
```

```
## [1] 0.0075
```

We can see that our ADABoost model with a committee size of 100 has an error rate of 0.0075.