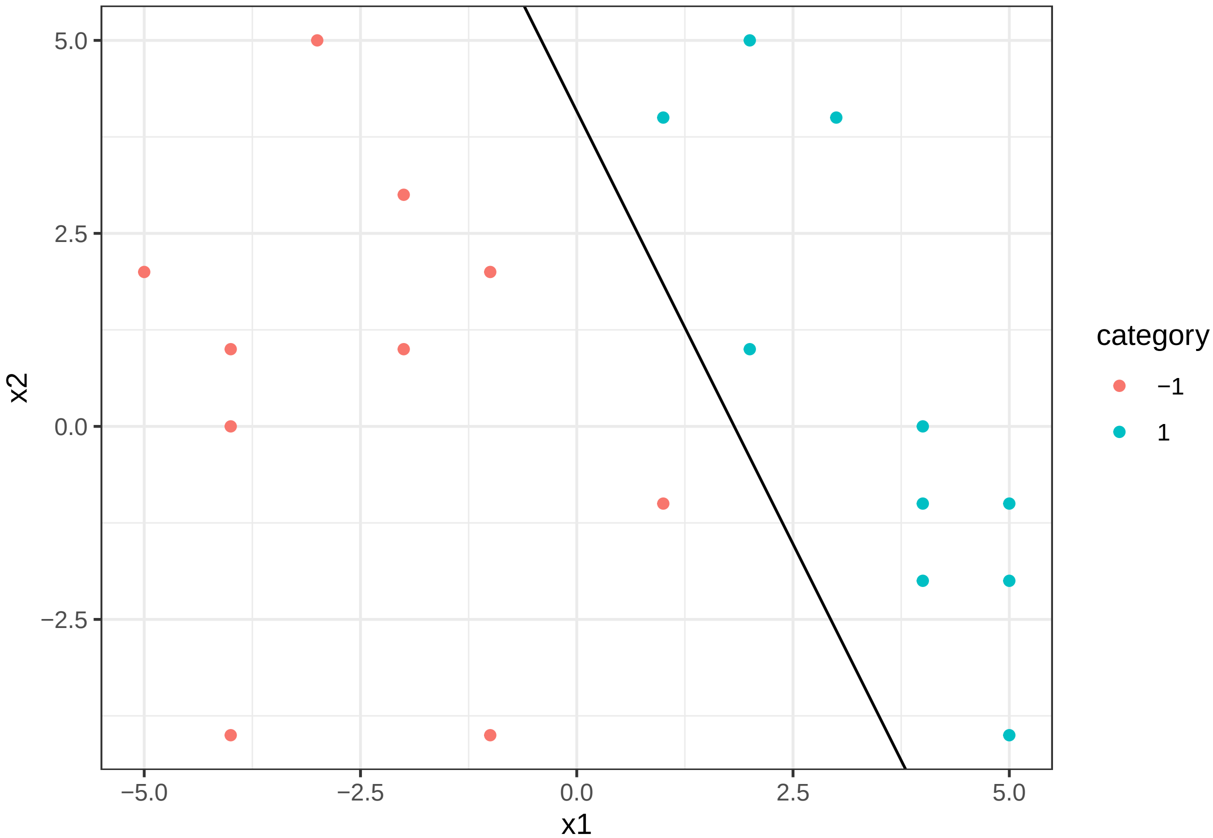
Lab3

Jia Lin Gao

25/08/2020

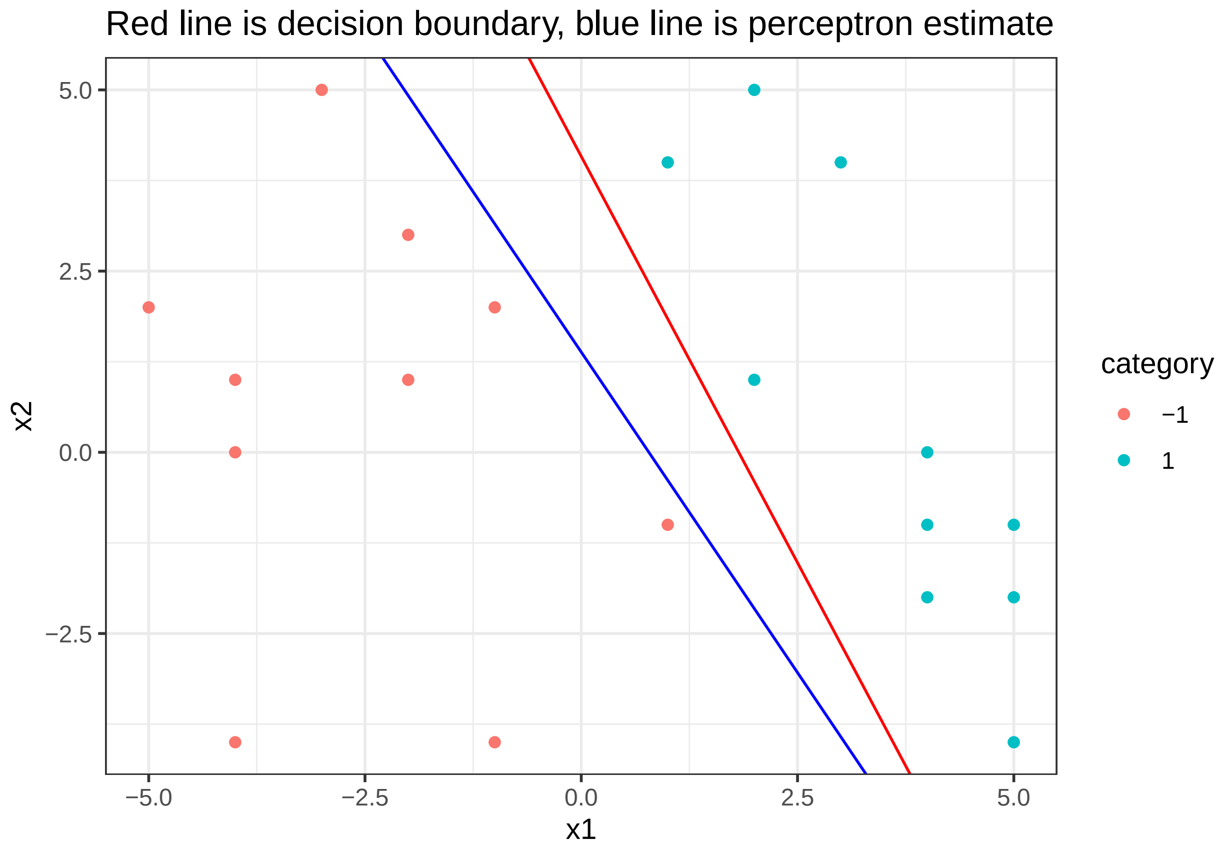
Contents

Exercise 1



The figure above shows an generated dataset D with $d = 2$ and $x_i^n \in R[-5, 5]$

```
## [1] "The perceptron learning algorithm took 6 updates to converge"
## [1] "This occurred in iteration 2"
```



The above output is the result of running a perceptron algorithm on the randomly generated dataset.

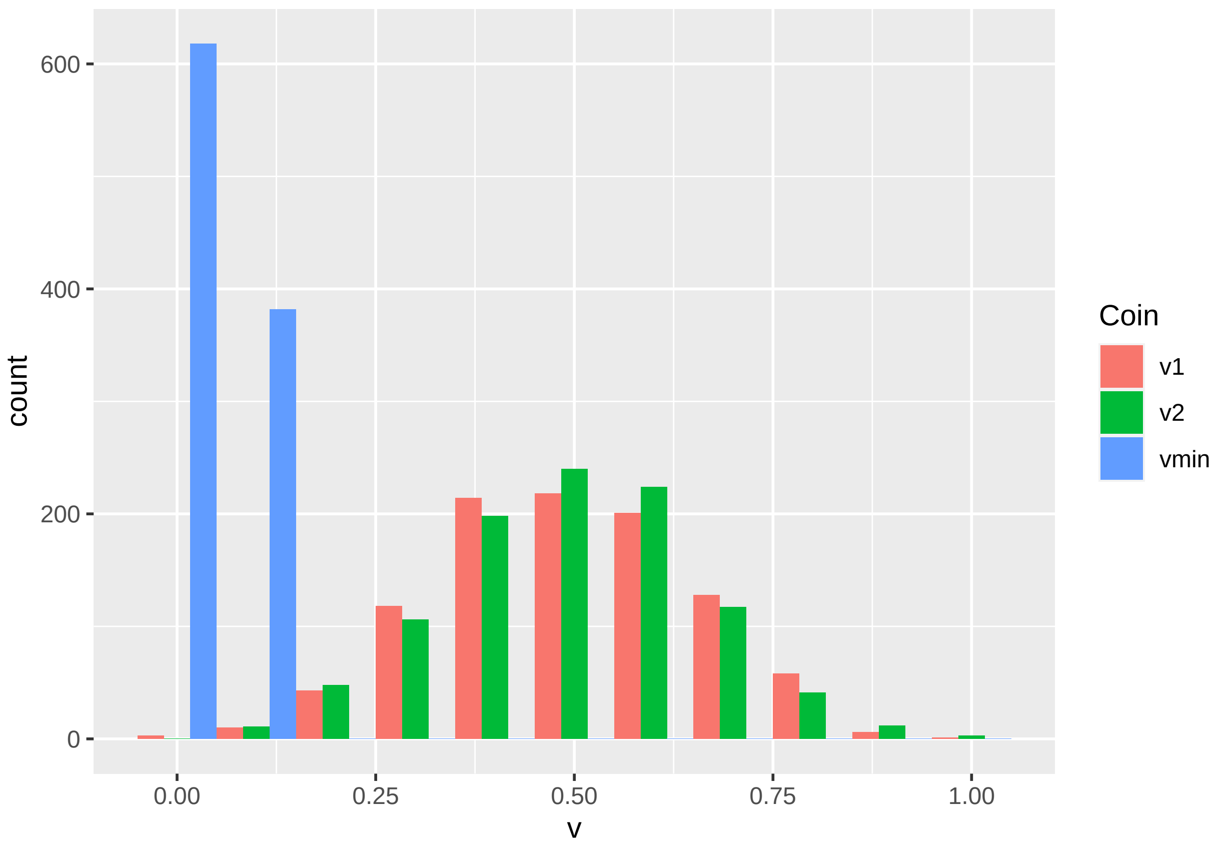
Exercise 2

Question 2A

The μ of any given fair (simulated) coin will be 0.5.

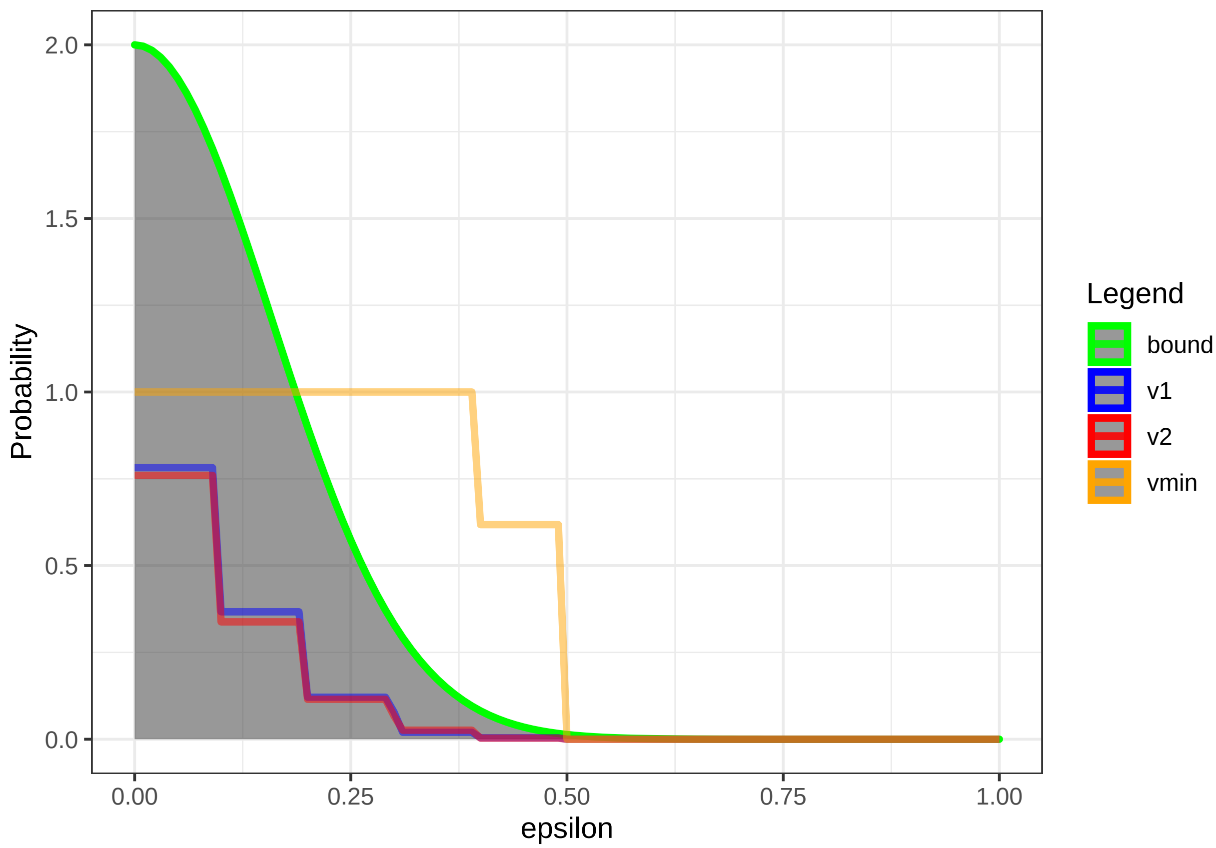
Question 2B

See histogram below



Question 2C

See the Hoeffding bound and the estimates for $Pr[|v - \mu| > \epsilon]$ plotted below



Question 2D

It appears that C_1 and C_2 obey the Hoeffding inequality, while C_{min} does not. This is because while the inequality applies to any individual bin before the sample data have been drawn, that is, we pick the random index of C_2 or say we will examine the first coin C_1 before drawing the samples. We can only examine C_{min} after sampling the data, in effect, we are being picky about choosing a sample that best fits a certain hypothesis. If we were to test the hypothesis that $Pr[|v - \mu| > \epsilon]$ is small for all of 1000 coins, we must also modify the bound from $2e^{-2\epsilon^2 N}$ to $2Me^{-2\epsilon^2 N}$ where M is your number of hypotheses, in this case, 1000.

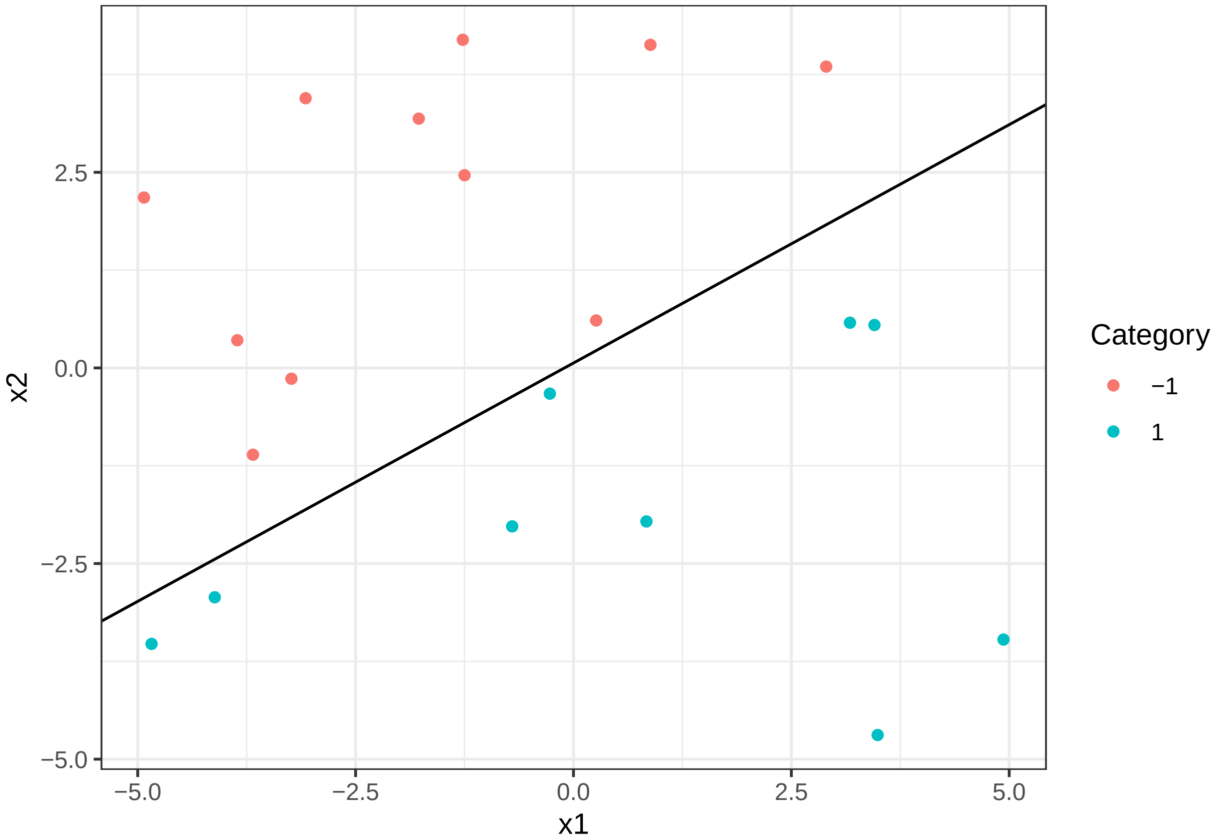
Question 2E

The experiment of flipping a coin is analogous to the act of drawing red and green marbles from a population, they are both binary outcome games. Drawing 1000 samples of 10 marbles from a jar that has equal amounts of red and green marbles will also have a ~67% chance of yielding a sample of only green marbles.

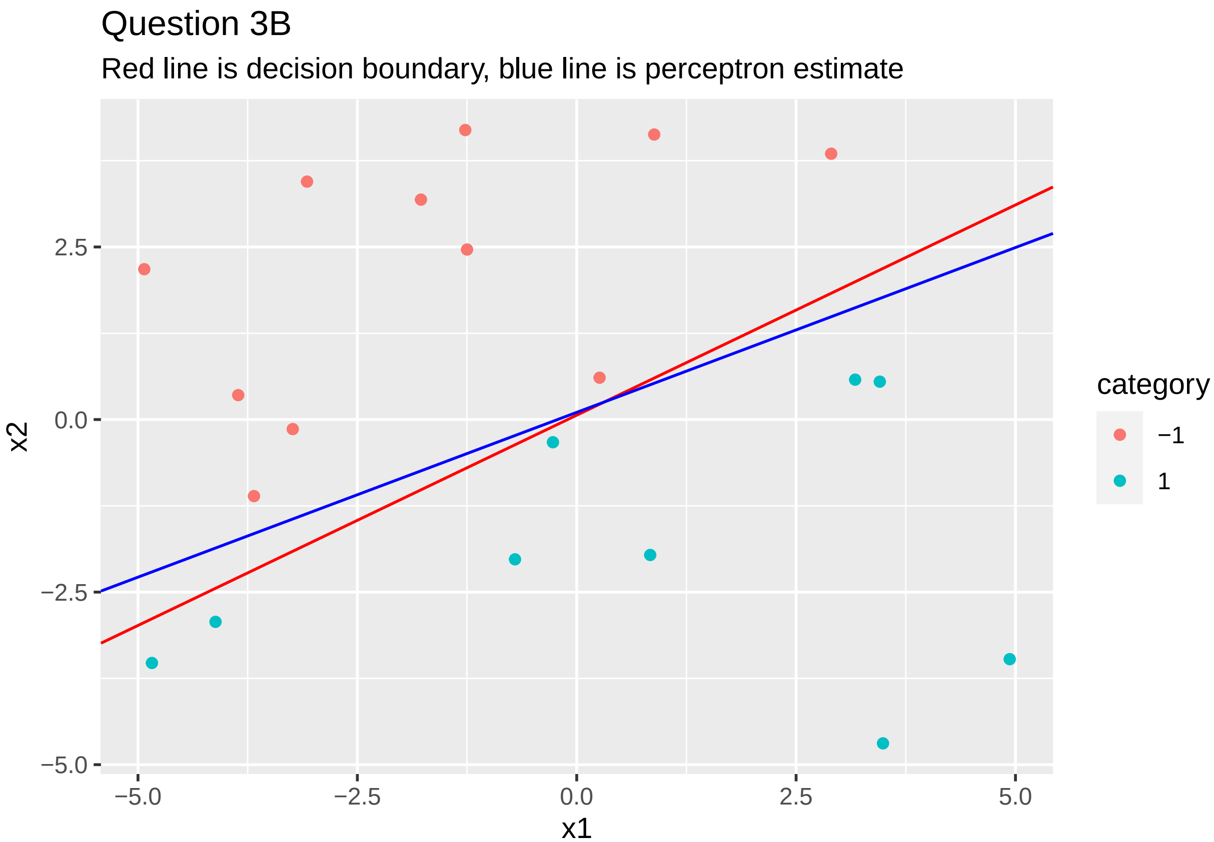
Exercise 3

Functions have been defined to generate the dataset for this exercise, to run the perceptron algorithm, as well as plot the results (code to animate the graphs have been commented out, but the animated .gif files are attached)

Question 3A



Question 3B

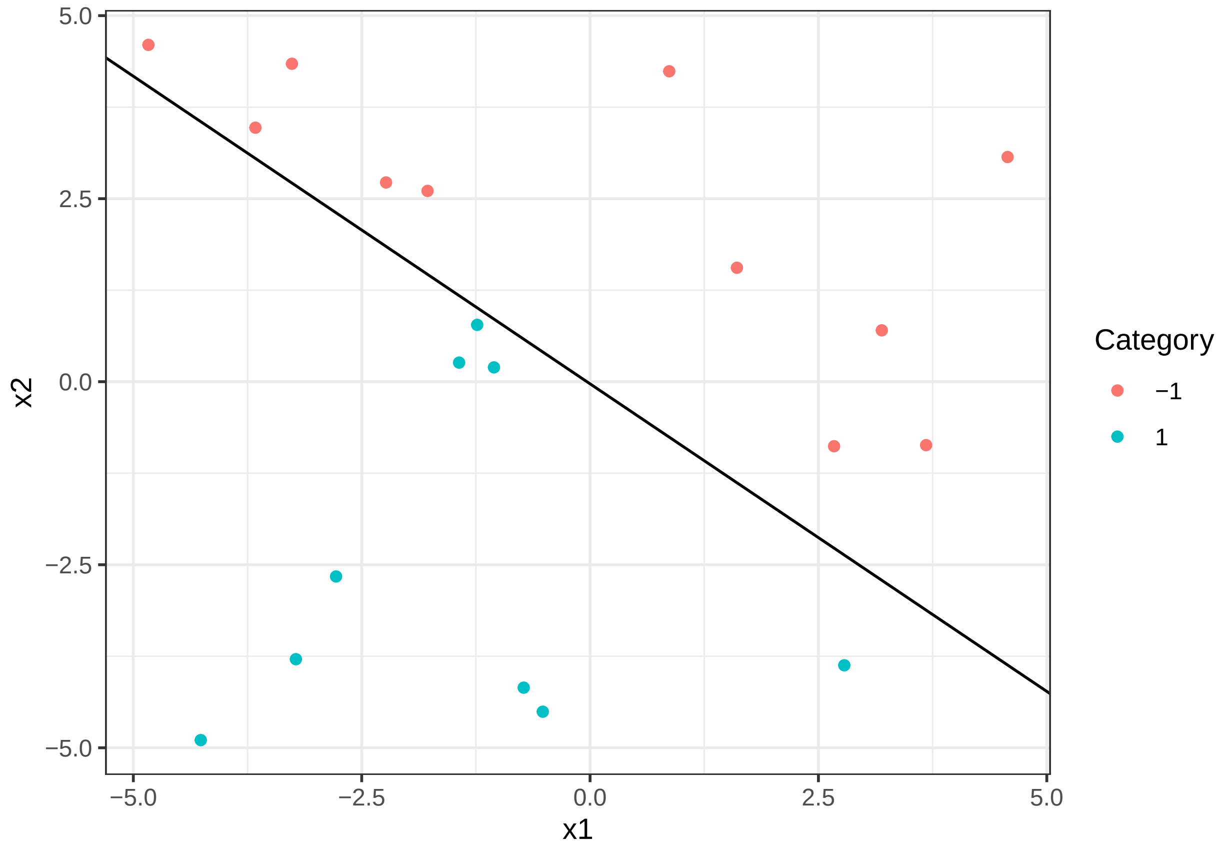


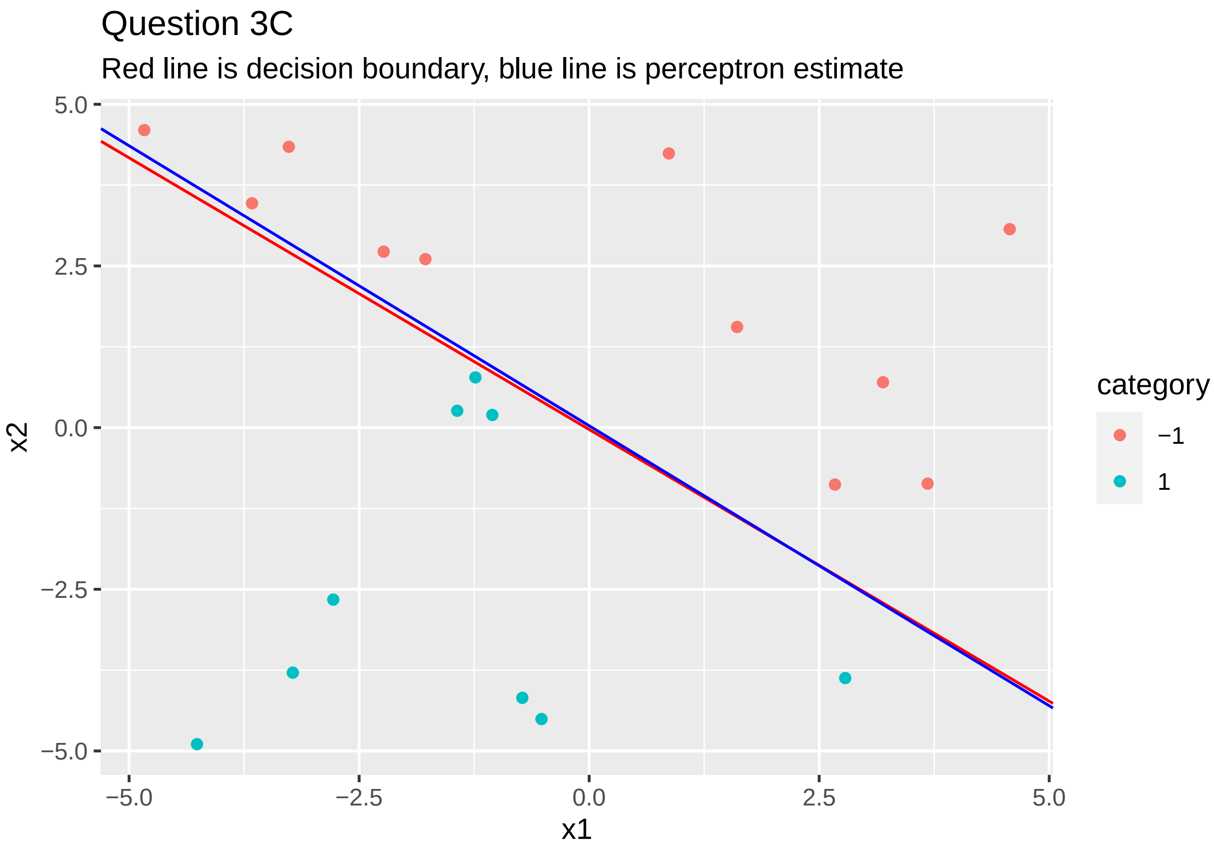

```
## [1] "The perceptron learning algorithm took 6 updates to converge"
## [1] "this occurred in iteration 2"
```

The estimated function g is quite close to the target function f

Question 3C

The dataset with a length of 20 data points is generated and a perceptron model is trained on it with the results shown below, the number of iterations taken should be similar to the results in Question 3B.

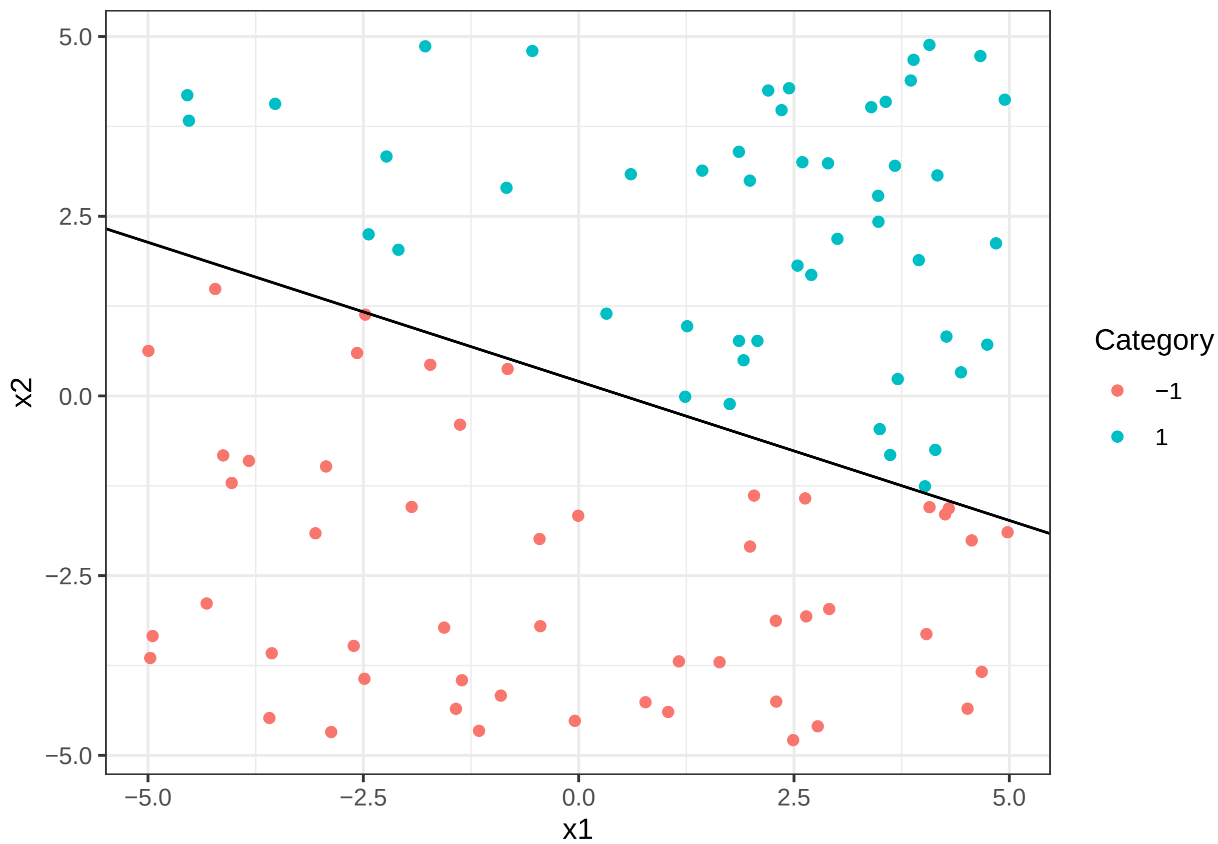

5.0
2.5
0.0
−2.5
−5.0
x2
−5.0
−2.5
0.0
2.5
5.0
x1
Category
−1
1

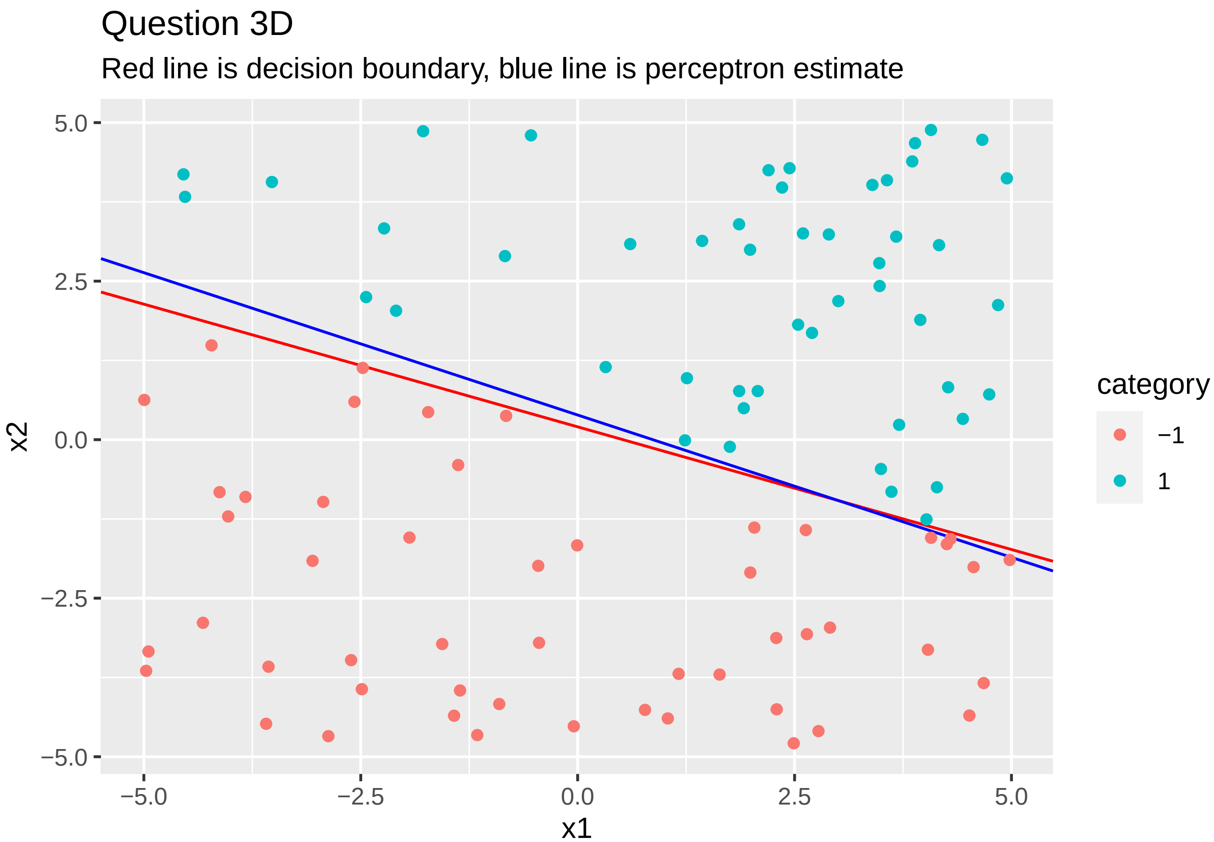



```
## [1] "The perceptron learning algorithm took 13 updates to converge"
## [1] "this occurred in iteration 3"
```

Question 3D

The dataset with a length of 100 data points is generated and a perceptron model is trained on it with the results shown below, the model required more updates to obtain the results from Question 3B.


5.0
2.5
0.0
−2.5
−5.0
x2
−5.0
−2.5
0.0
2.5
5.0
x1
Category
−1
1



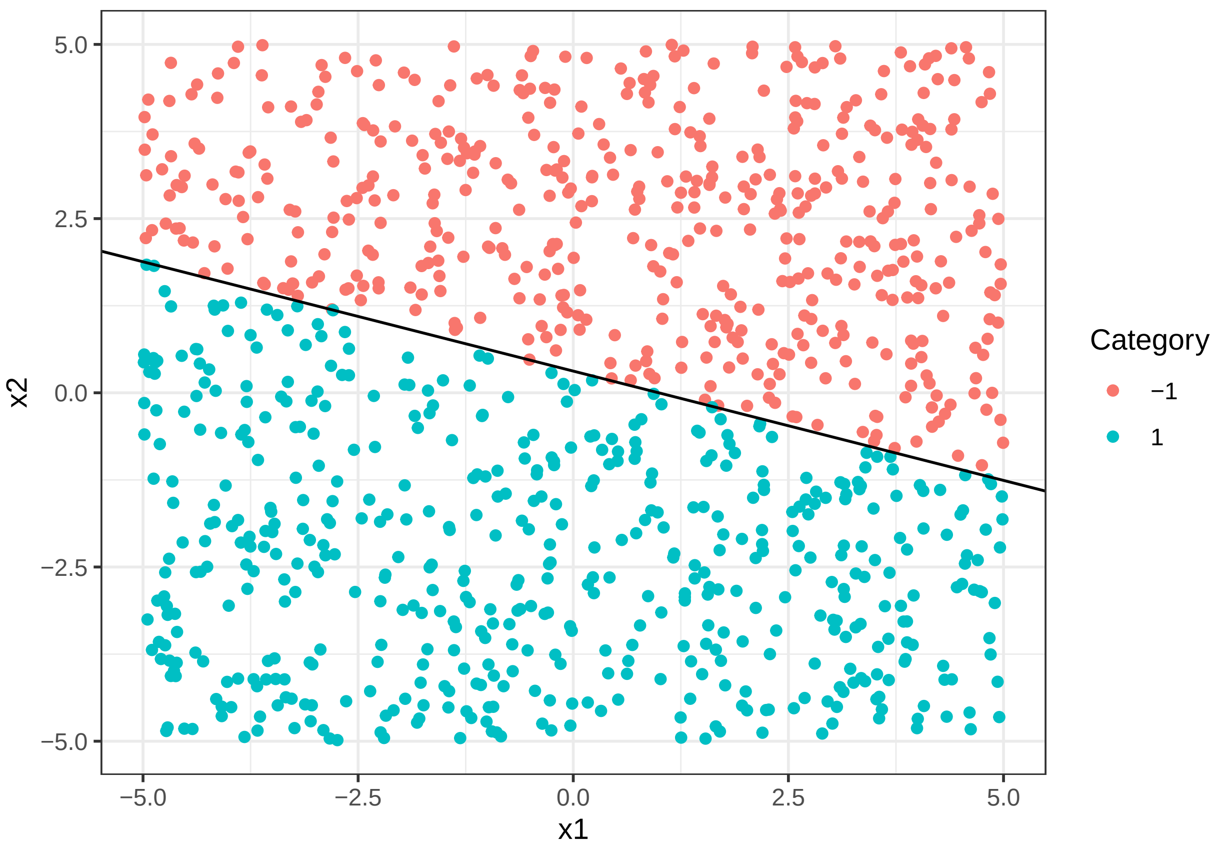

```
## [1] "The perceptron learning algorithm took 33 updates to converge"

## [1] "this occurred in iteration 4"
```

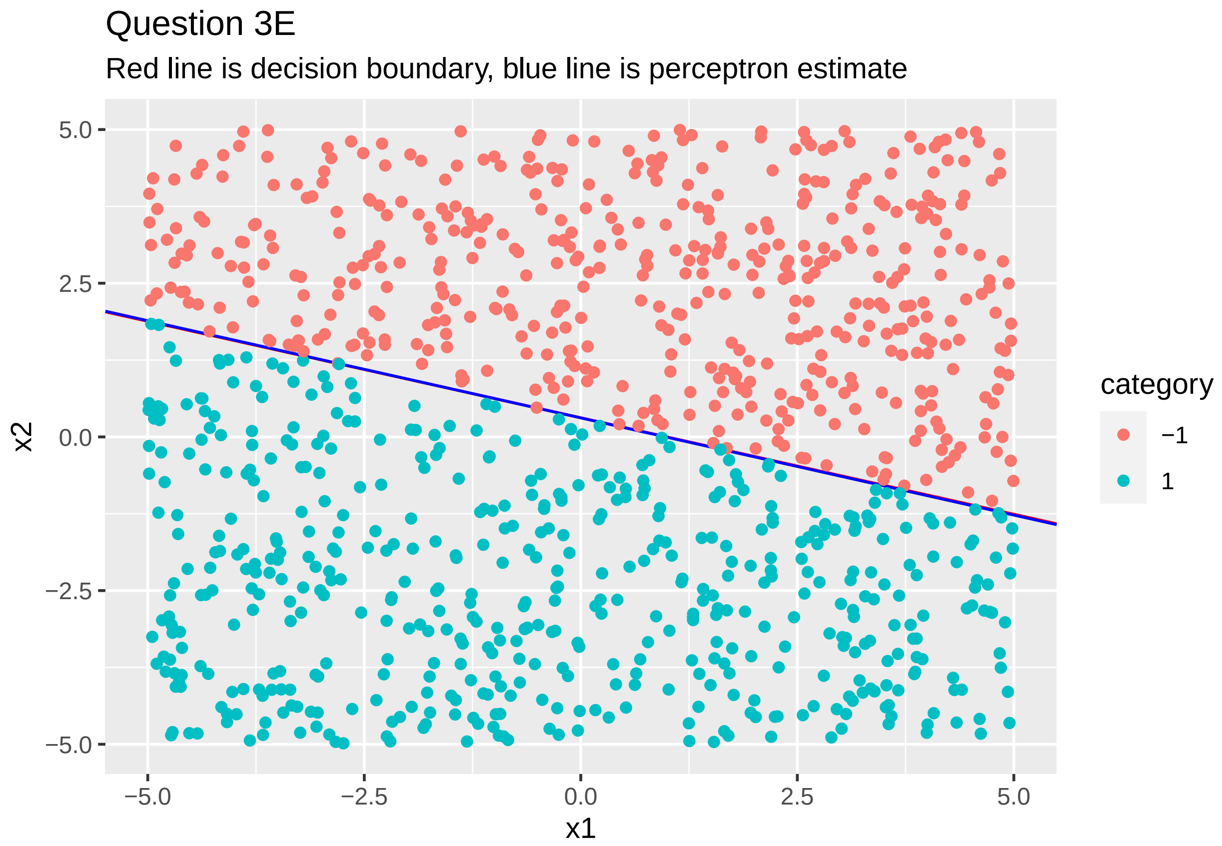
Question 3E

The dataset with a length of 1000 data points is generated and a perceptron model is trained on it with the results shown below, the model required more updates to obtain the results from Question 3B, 3C and 3E. However the increase in the number of updates required is proportionally less than the increase in dataset length.

We observe that the larger the dataset, the closer the perceptron decision boundary is to the original.




```
## [1] "The perceptron learning algorithm took 228 updates to converge"
## [1] "this occurred in iteration 21"
```



Question 3F

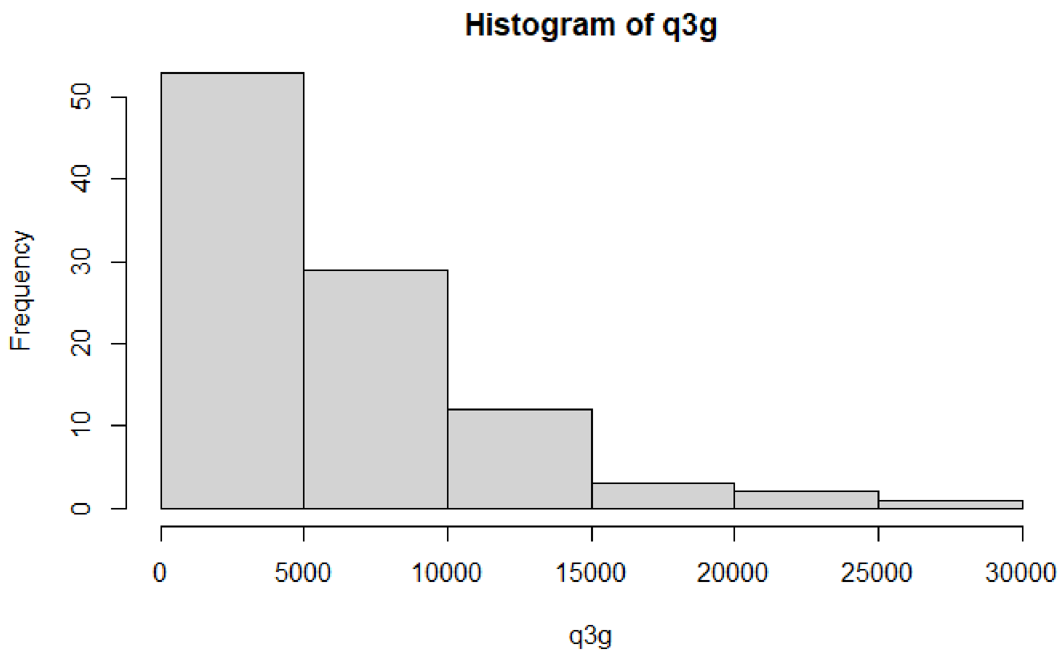
The results of the algorithm applied to a randomly generated dataset with 10 dimensions is shown below, the increase in dimensions has has a far greater effect on updates required before convergence.

```
## [1] "The perceptron learning algorithm took 2267 updates to converge"
```

```
## [1] "This occurred in iteration 216"
```

Question 3G

The histogram that is the result of observing the iterations taken when Question 3F is repeated with new random datasets is shown below



Question 3H

The questions appear to infer that convergence is required, therefore accuracy of the classifier would be identical for all the models evaluated in exercise 3. However, increasing d has a far greater effect in increasing the number of updates required for the perceptron algorithm to converge on a working decision boundary compared to increasing N.