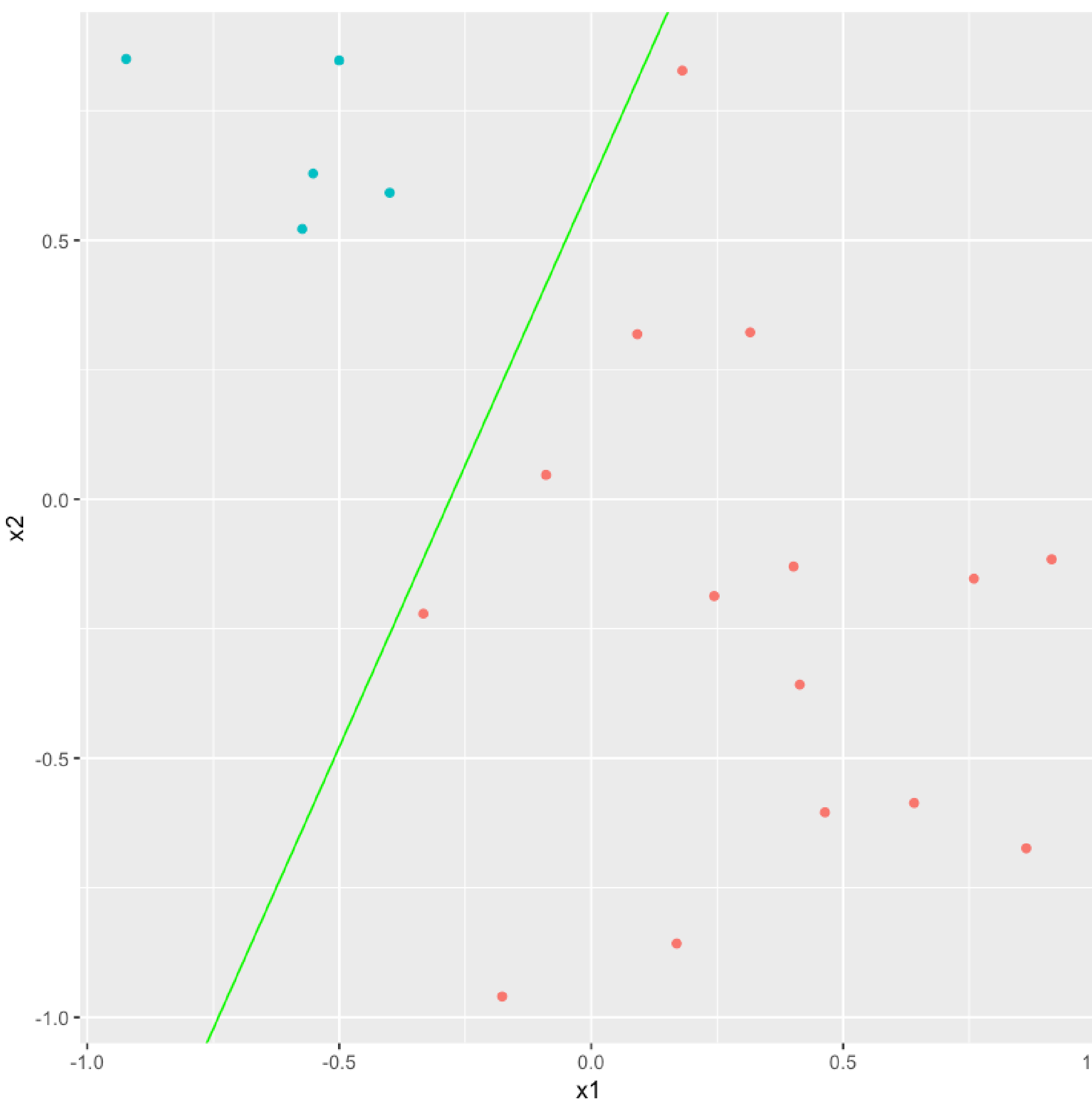
In [ ]:
```
NOTE : The below code has been written in R kindly install R packages in
 jupyter notebook before running them.
```

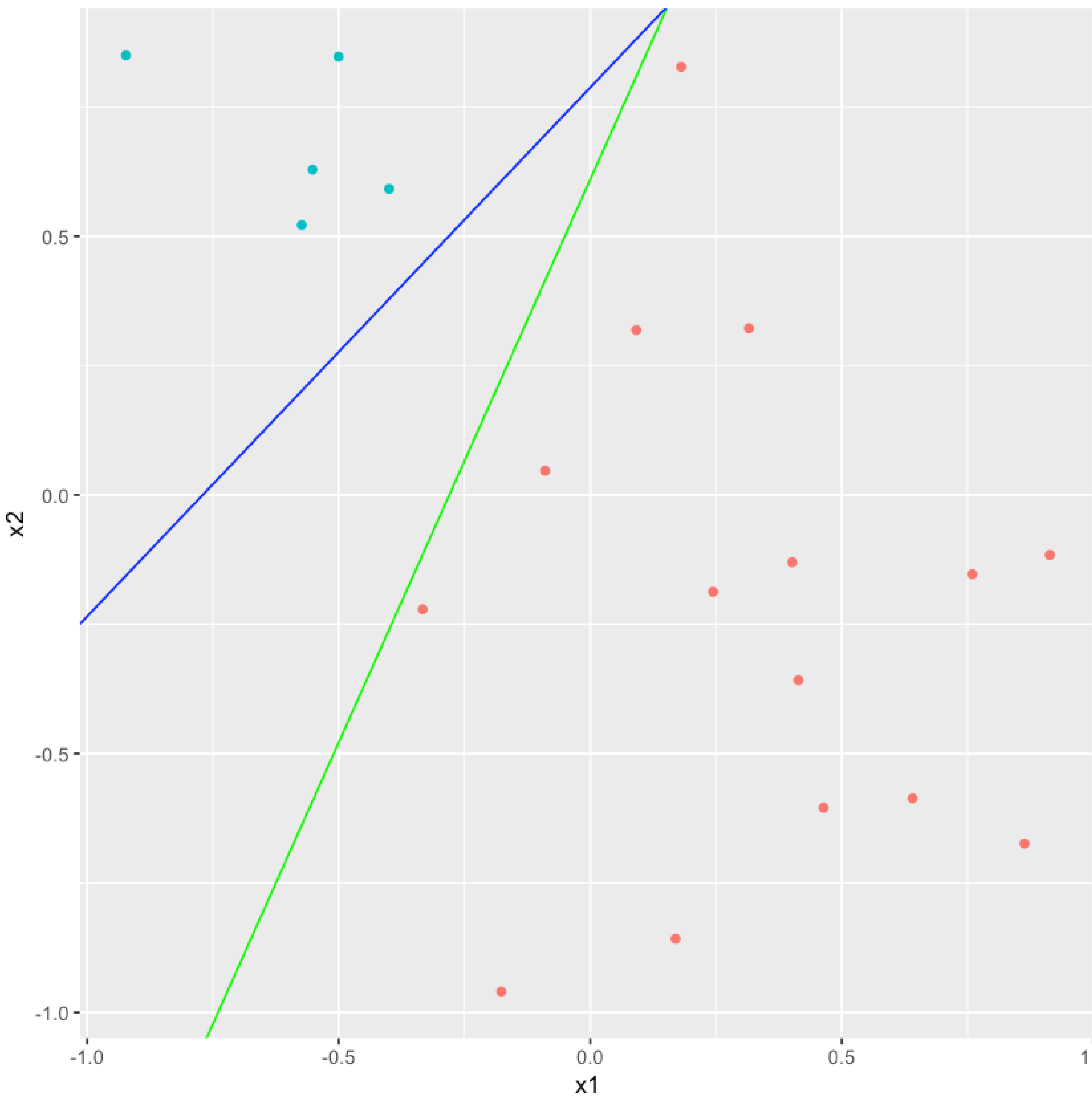
In [ ]:
```
(a) Below, we generate a linearly separable data set of size 20 and the
 target function f (in red)
```

In [13]:
```
library(ggplot2)
set.seed(101)
h <- function(x, w) {
scalar_product <- cbind(1, x$x1, x$x2) %*% w
    return(as.vector(sign(scalar_product)))
}
w0 <- runif(1, min = -999, max = 999)
w1 <- runif(1, min = -999, max = 999)
w2 <- runif(1, min = -999, max = 999)
f <- function(x) {
    return(h(x, c(w0, w1, w2)))
}
D <- data.frame(x1 = runif(20, min = -1, max = 1), x2 = runif(20, min =
-1, max = 1))
D <- cbind(D, y = f(D))
dots <- ggplot(D, aes(x = x1, y = x2, col = as.factor(y + 3))) + geom_po
int() + theme(legend.position = "none")
p_f <- dots + geom_abline(slope = -w1 / w2, intercept = -w0 / w2, colour
 = "green")
p_f
```



In [ ]:
```
(b) Below, we plot the training data,
the target function f (in red) and the final hypothesis g (in blue) gene
rated by PLA
```

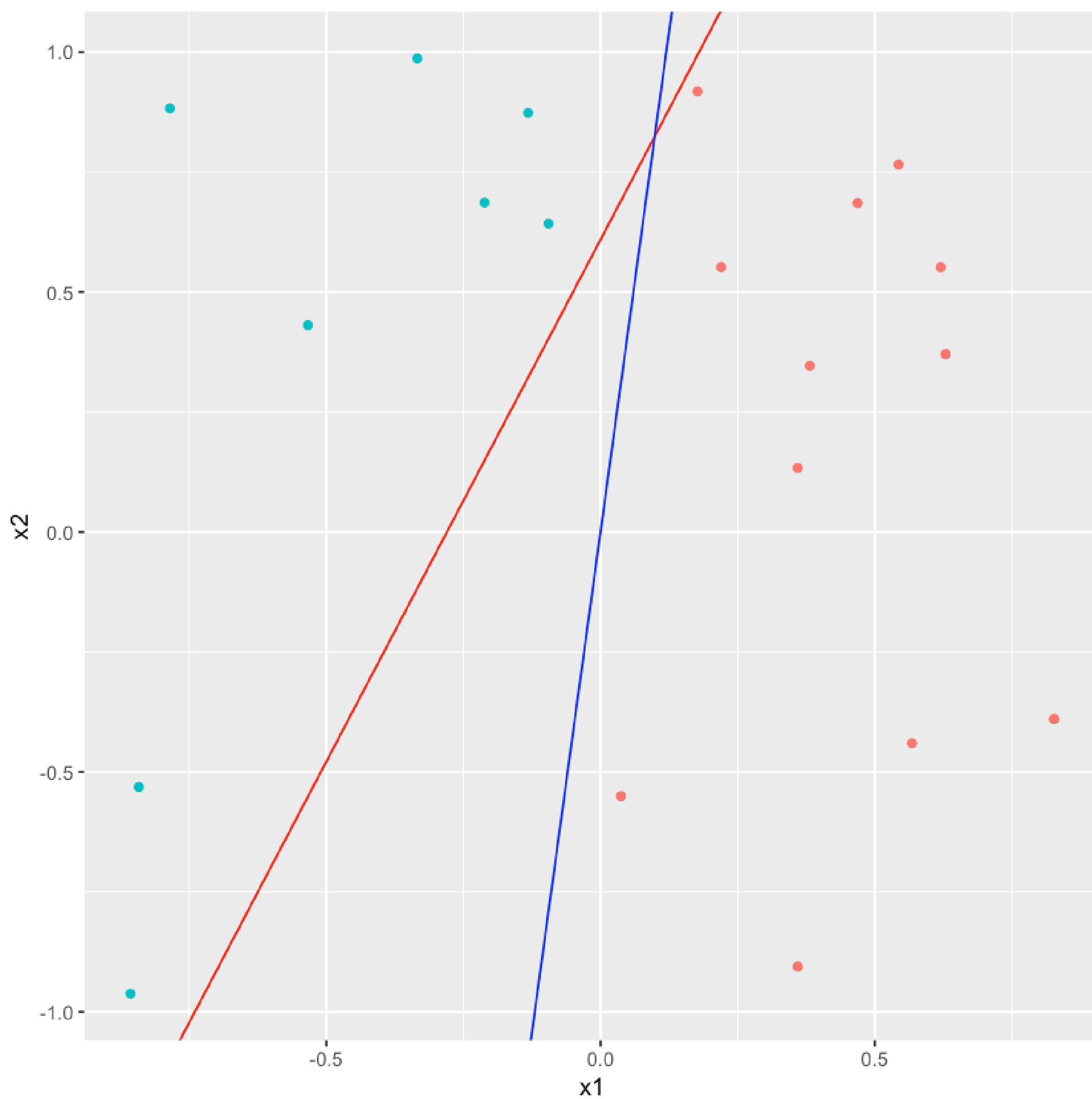
In [9]:
```
iter <- 0
w <- c(0, 0, 0)
repeat {
y_prediction <- h(D, w)
D_mis <- subset(D, y != y_prediction)
    if (nrow(D_mis) == 0)
break
x_t <- D_mis[1, ]
w <- w + c(1, x_t$x1, x_t$x2) * x_t$y
    iter <- iter + 1
}
p_g <- p_f + geom_abline(slope = -w[2] / w[3], intercept = -w[1] / w[3],
 colour = "blue")
p_g
```



In [ ]:
```
Here, the PLA took 5 iterations before converging.
We may notice that although g is close to f,but are not converging
```

In [ ]:
```
(c) Below, we repeat what we did in point (b) with another randomly gene
rated data set of size 20
```

In [10]:
```
D1 <- data.frame(x1 = runif(20, min = -1, max = 1), x2 = runif(20, min =
 -1, max = 1))
D1 <- cbind(D1, y = f(D1))
iter <- 0
w <- c(0, 0, 0)
repeat {
y_prediction <- h(D1, w)
D_mis <- subset(D1, y != y_prediction)
    if (nrow(D_mis) == 0)
break
x_t <- D_mis[1, ]
w <- w + c(1, x_t$x1, x_t$x2) * x_t$y
    iter <- iter + 1
}
ggplot(D1, aes(x = x1, y = x2, col = as.factor(y + 3))) + geom_point() +
 theme(legend.position = "none") +
geom_abline(slope = -w1 / w2, intercept = -w0 / w2, colour = "red") + ge
om_abline(slope = -w[2] / w[3], intercept = -w[1] / w[3], colour = "blu
e")
```

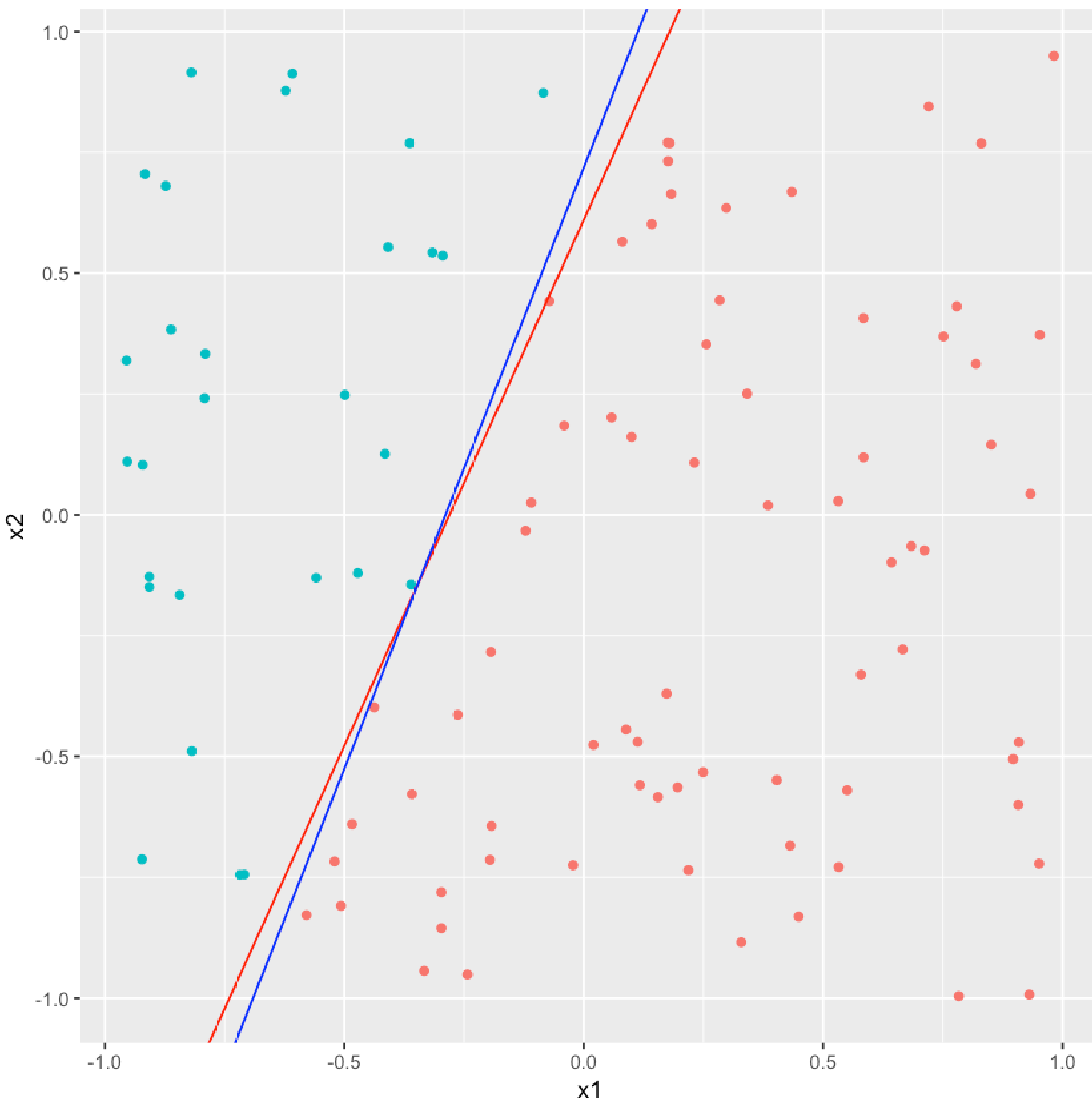


In [ ]:
```
In this case, the PLA took 12 iterations (which is greater than in (b))
 before converging.
We may notice that, as in point (b), although g is pretty close to f,
they are not quite identical
```

In [ ]:
```
(d) Below, we repeat what we did in point (b) with another randomly gene
rated data set of size 100.
```

In [12]:

```
D1 <- data.frame(x1 = runif(100, min = -1, max = 1), x2 = runif(100, min
 = -1, max = 1))
D1 <- cbind(D1, y = f(D1))
iter <- 0
w <- c(0, 0, 0)
repeat {
y_prediction <- h(D1, w)
D_mis <- subset(D1, y != y_prediction)
    if (nrow(D_mis) == 0)
break
x_t <- D_mis[1, ]
w <- w + c(1, x_t$x1, x_t$x2) * x_t$y
    iter <- iter + 1
}
ggplot(D1, aes(x = x1, y = x2, col = as.factor(y + 3))) + geom_point() +
 theme(legend.position = "none") +
geom_abline(slope = -w1 / w2, intercept = -w0 / w2, colour = "red") + ge
om_abline(slope = -w[2] / w[3], intercept = -w[1] / w[3], colour = "blu
e")
```



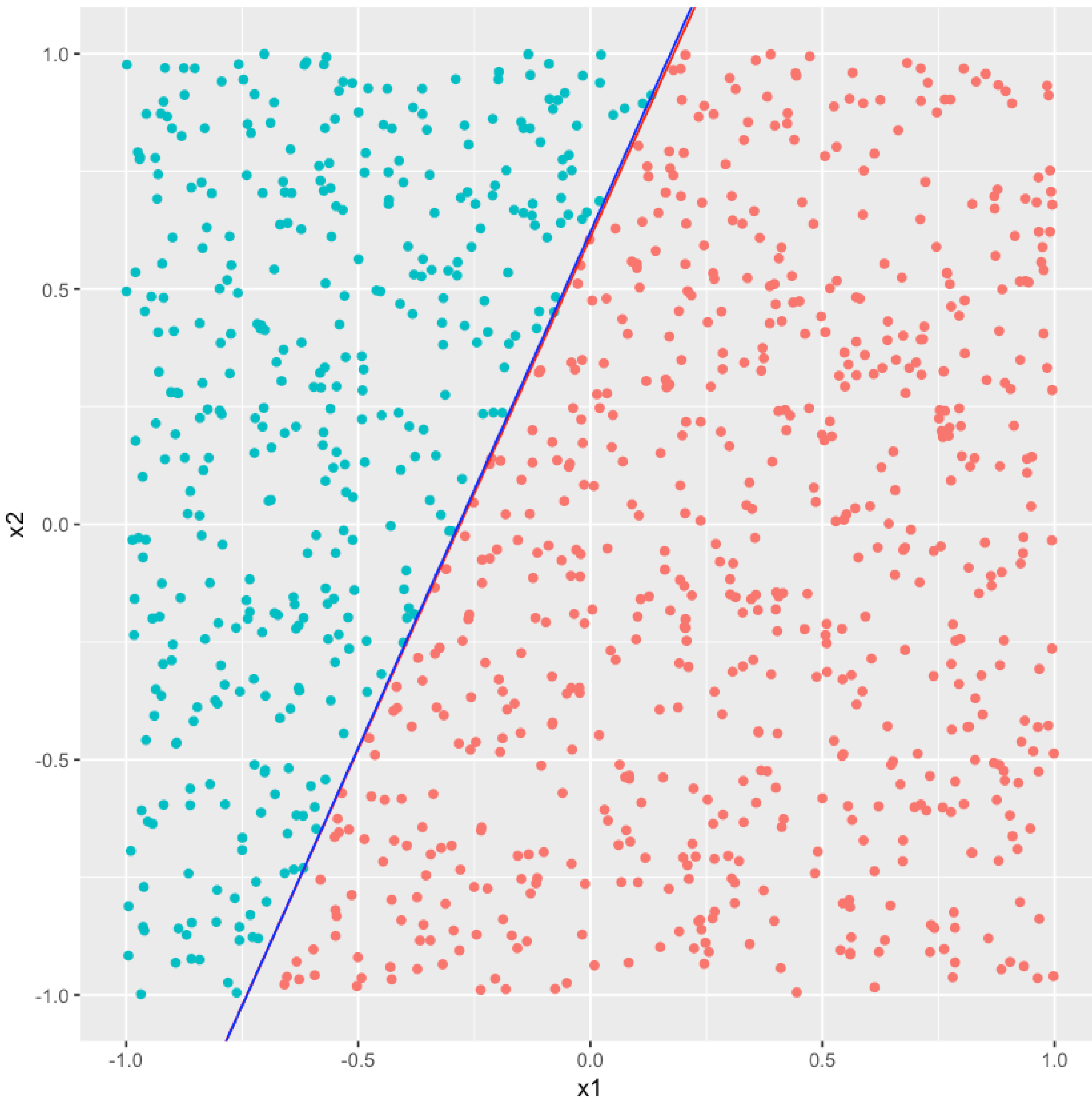
In [ ]:

```
In this case, the PLA took 33 iterations (which is greater than in (b) a
nd (c)) before converging.
We may notice that, here f and g are very close to each other
```

In [ ]:

```
e) Below, we repeat what we did in point (b) with another randomly gener
ated data set of size 1000
```

In [11]:
```
D1 <- data.frame(x1 = runif(1000, min = -1, max = 1), x2 = runif(1000, m
in = -1, max = 1))
D1 <- cbind(D1, y = f(D1))
iter <- 0
w <- c(0, 0, 0)
repeat {
y_prediction <- h(D1, w)
D_mis <- subset(D1, y != y_prediction)
    if (nrow(D_mis) == 0)
break
x_t <- D_mis[1, ]
w <- w + c(1, x_t$x1, x_t$x2) * x_t$y
    iter <- iter + 1
}
ggplot(D1, aes(x = x1, y = x2, col = as.factor(y + 3))) + geom_point() +
 theme(legend.position = "none") +
geom_abline(slope = -w1 / w2, intercept = -w0 / w2, colour = "red") + ge
om_abline(slope = -w[2] / w[3], intercept = -w[1] / w[3], colour = "blu
e")
```



In [ ]:
```
In this case, the PLA took 511 iterations (which is greater than in (b),
 (c) and (d)) before converging.
We may notice that, here f and g are nearly undistinguishable.
```

In [ ]:
```
(f) Here, we randomly generate a linearly separable data set of size 100
0 with xn ∈ R10.
```

In [20]:
```
N <- 10
h <- function(x, w) {
scalar_product <- cbind(1, x) %*% w
return(as.vector(sign(scalar_product))) }
w <- runif(N + 1)
f <- function(x)
{
    return(h(x, w))
}
D2 <- matrix(runif(10000, min = -1, max = 1), ncol = N)
D2 <- cbind(D2, y = f(D2))
D2 <- data.frame(D2)
iter <- 0
w0 <- rep(0, N + 1)
repeat {
y_pred <- h(as.matrix(D2[, 1:N]), as.numeric(w0))
    D_mis <- subset(D2, y != y_pred)
if (nrow(D_mis) == 0)
break
x_t <- D_mis[1, ]
w0 <- w0 + cbind(1, x_t[, 1:N]) * x_t$y
    iter <- iter + 1
}
```