Comparing LDA and Logistic Testing Misclassification Rates

```
library(ISLR)
data(Default)
head(Default)
```

```
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

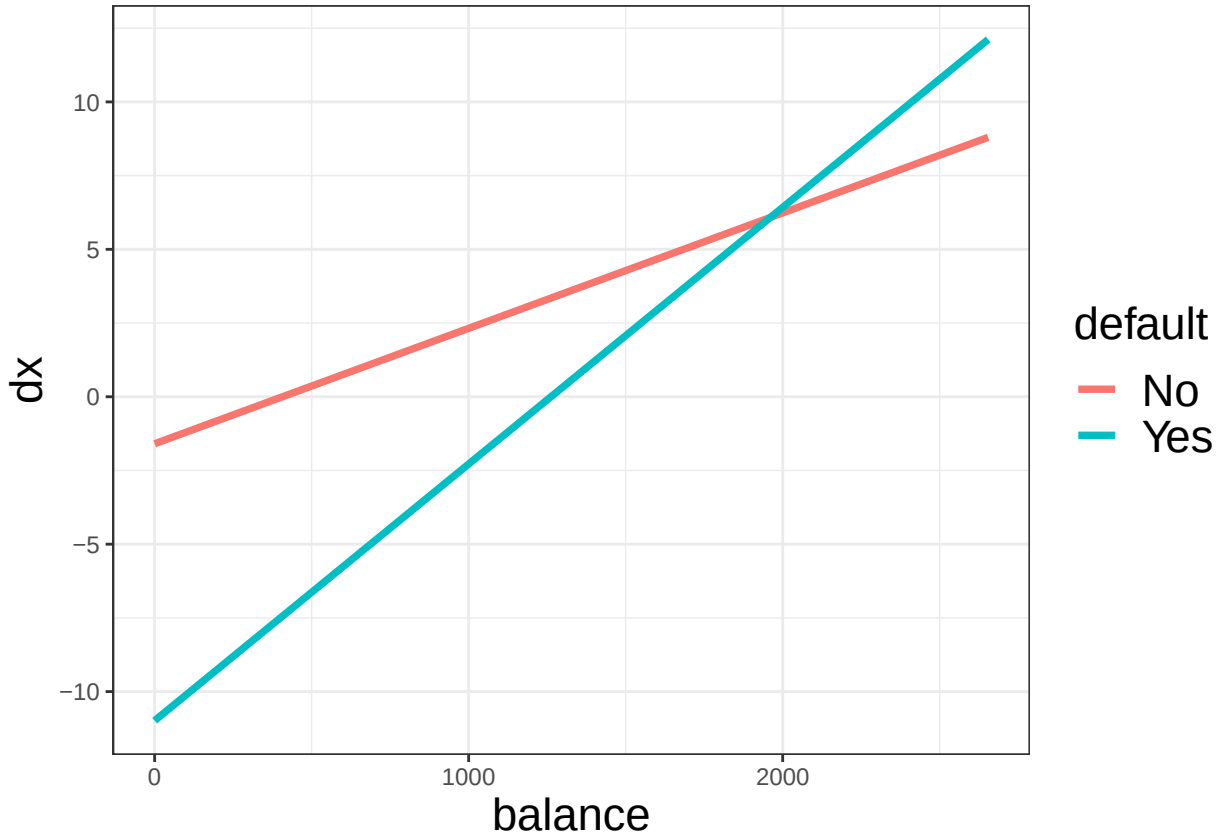
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(39)
test_ind <- sample(1:10000, size = 5000)
Default_test <- Default[test_ind, ]
Default_train <- Default[-test_ind, ]

est <- Default_train %>%
  group_by(default) %>%
  summarize(n = n(),
            prop = n/nrow(Default_train),
            mu = mean(balance),
            ssx = var(balance) * (n - 1))
est
```

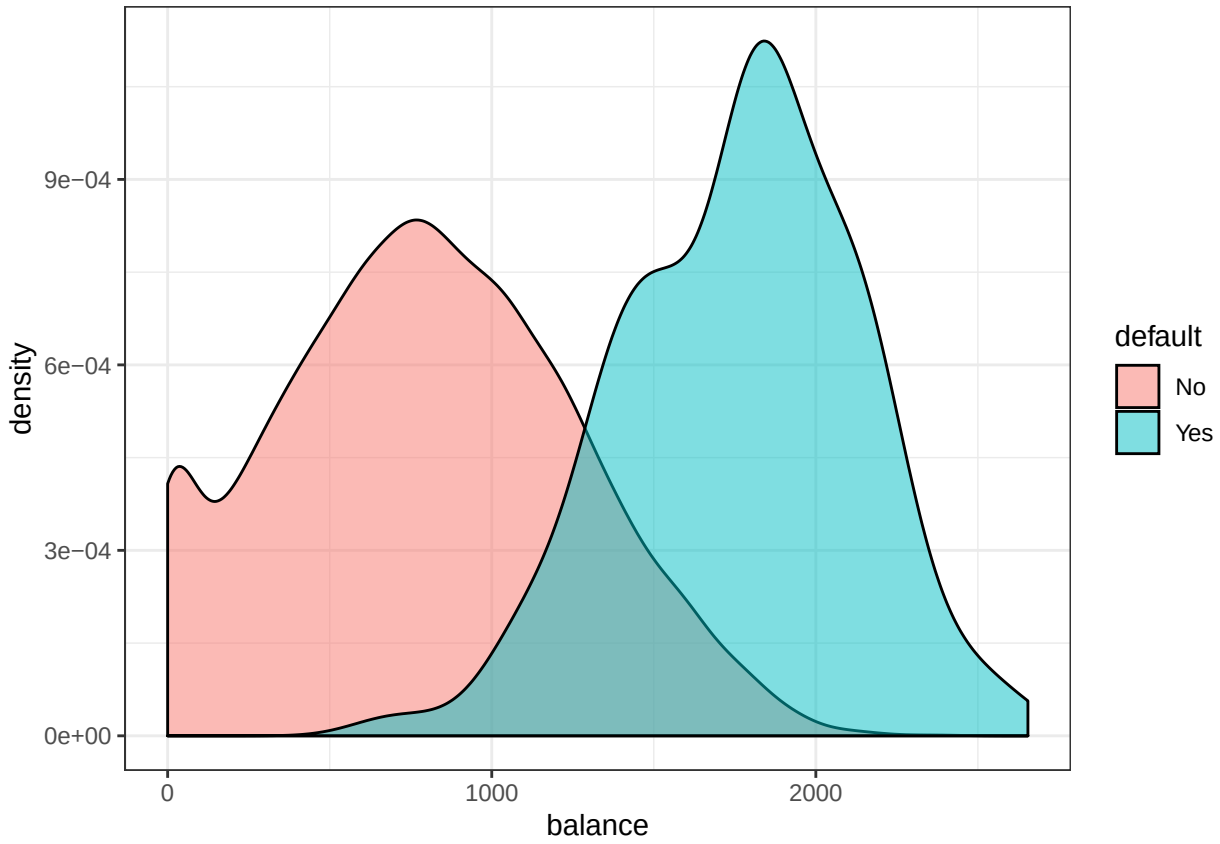
```
## # A tibble: 2 x 5
##   default     n   prop    mu        ssx
##   <fct>   <int>  <dbl> <dbl>      <dbl>
## 1 No       4814 0.963   795. 991334919.
## 2 Yes       186 0.0372 1768.  23803633.
```

```
pi_n <- as.numeric(est[1, 3])
pi_y <- as.numeric(est[2, 3])
mu_n <- as.numeric(est[1, 4])
mu_y <- as.numeric(est[2, 4])
sig_sq <- (1/(nrow(Default_train) - 2)) * sum(est$ssx)

my_lda <- function(x, pi, mu, sig_sq) {
  x * (mu/sig_sq) - (mu^2)/(2 * sig_sq) + log(pi)
```

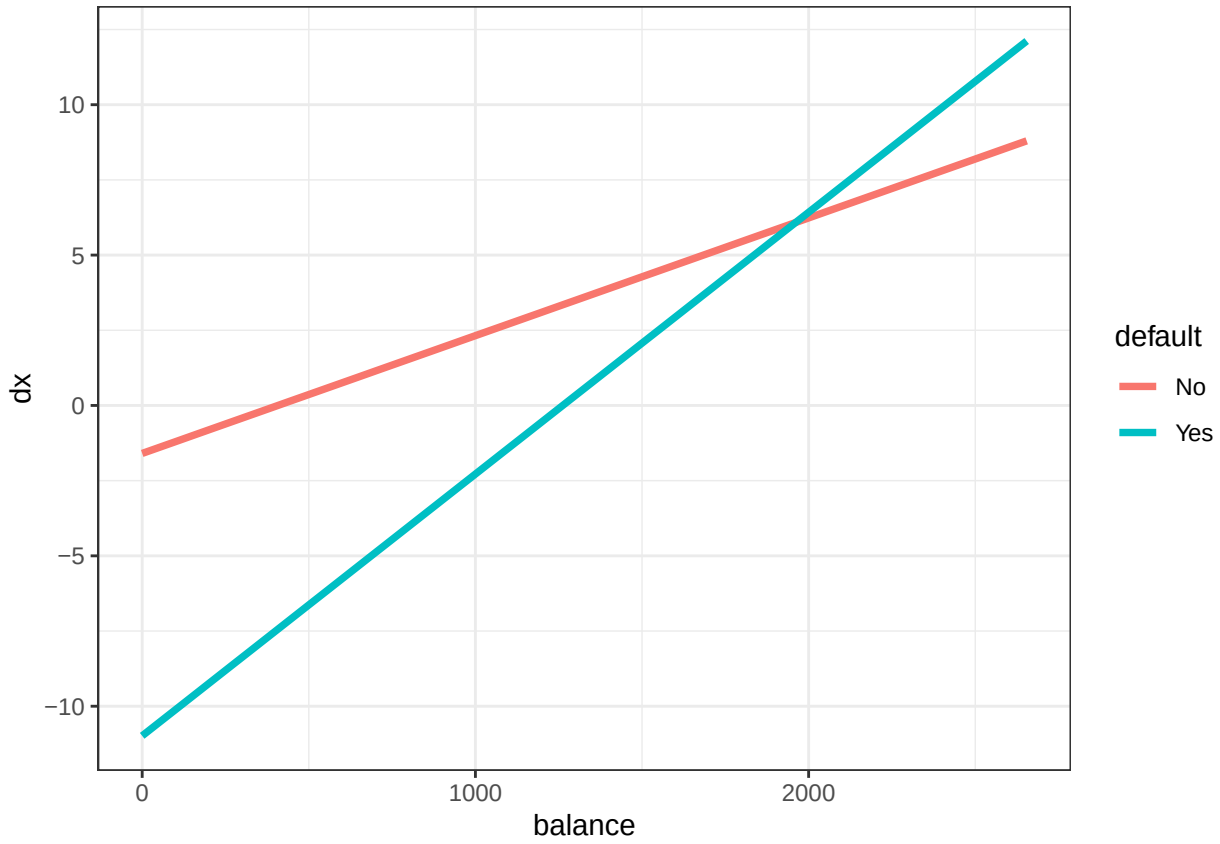
```
}

d_n <- my_lda(Default_train$balance, pi_n, mu_n, sig_sq)
d_y <- my_lda(Default_train$balance, pi_y, mu_y, sig_sq)


## ---- echo = FALSE, fig.width=8, fig.height = 5.5, fig.align = "center"----
library(ggplot2)
balance <- seq(0, max(Default_train$balance), length.out = 50)
dx <- c(my_lda(balance, pi_n, mu_n, sig_sq), my_lda(balance, pi_y, mu_y, sig_sq))
default <- as.factor(rep(c("No", "Yes"), each = 50))
df <- data.frame(balance = rep(balance, 2), dx, default)

p1 <- ggplot(df, aes(x = balance, y = dx, color = default)) +
  geom_line(lwd = 1.3) +
  theme_bw()
p1 +
  theme(axis.title.x = element_text(size = rel(1.5)),
        axis.title.y = element_text(size = rel(1.5)),
        legend.text  = element_text(size = rel(1.5)),
        legend.title = element_text(size = rel(1.5)))
```




```
## ----echo = FALSE, fig.align="center", fig.height = 2.5-------------------
Default_train %>%
  ggplot(aes(x = balance, fill = default)) +
  geom_density(alpha = .5) +
  theme_bw()
```

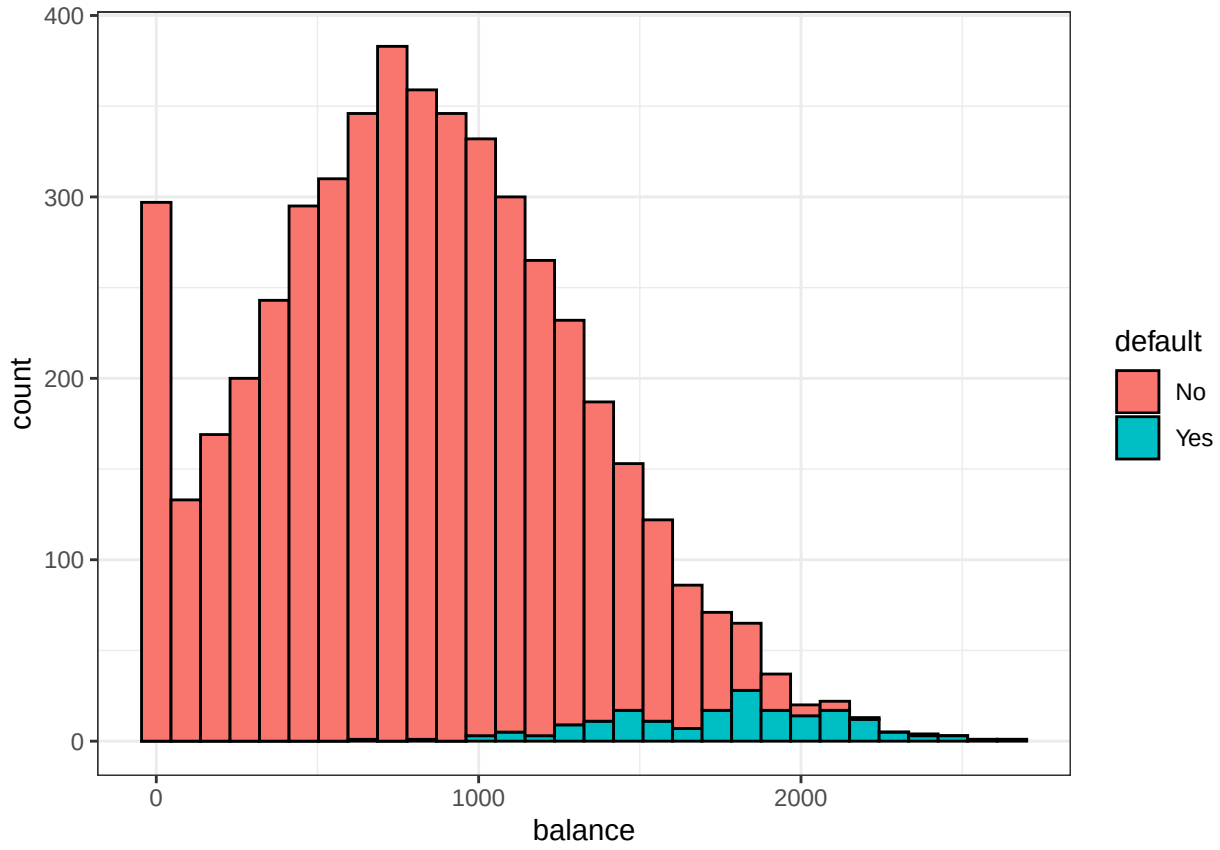

9e−04
6e−04
3e−04
0e+00
density
0
1000
2000
balance
default
No
Yes

```
p1
```

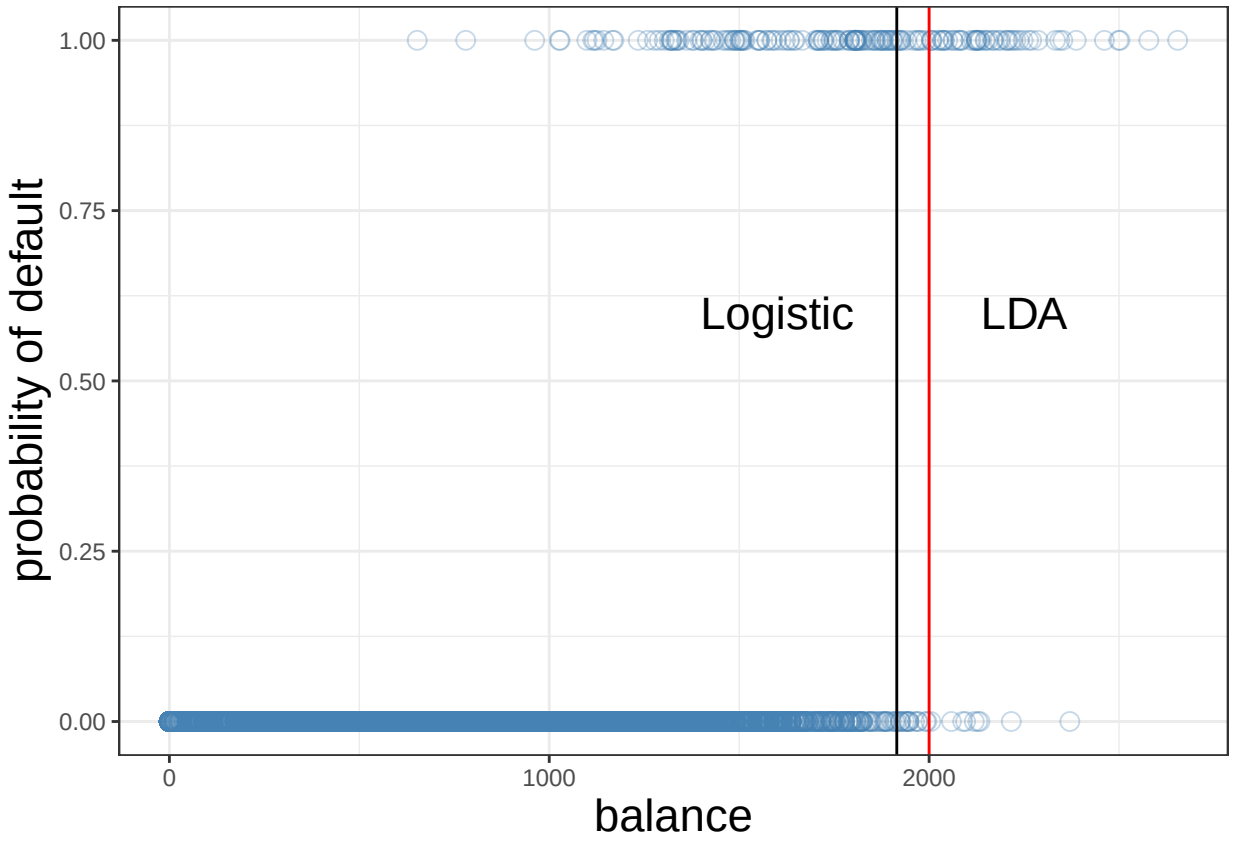



```
Default_train %>%
  ggplot(aes(x = balance, fill = default)) +
  geom_histogram(col = "black") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## ---- echo = FALSE, fig.align="center", fig.heigh = 8---------------------
Default_train <- mutate(Default_train, defaultYes = ifelse(default == "Yes", 1, 0))

p3 <- ggplot(Default_train, aes(x = balance, y = defaultYes)) +
  geom_point(pch = 1, alpha = .3, color = "steelblue", size = 3) +
  ylab("probability of default") +
  theme_bw()

m1 <- glm(default ~ balance, data = Default_train, family = binomial)
thresh <- (log(.5/.5) - m1$coef[1])/m1$coef[2]

p3 + geom_vline(xintercept = 2000, col = 2) +
  geom_vline(xintercept = thresh) +
  annotate(geom = "text", x = 1600, y = .6,
           label = "Logistic", size = 6) +
  annotate(geom = "text", x = 2250, y = .6,
           label = "LDA", size = 6) +
  theme(axis.title.x = element_text(size = rel(1.5)),
        axis.title.y = element_text(size = rel(1.5)))
```




```
## ------------------------------------------------------------------------
m1 <- glm(default ~ balance, data = Default_train,
          family = binomial)
my_log_pred <- ifelse(m1$fit < 0.5, "No", "Yes")
my_lda_pred <- ifelse(d_n > d_y, "No", "Yes")


## ----df, eval = FALSE----------------------------------------------------
## data.frame(log_pred = my_log_pred[8459:8464],
##            lda_pred = my_lda_pred[8459:8464],
##            true = Default$default[8459:8464])


## ----ref.label = "df", echo = FALSE--------------------------------------
data.frame(log_pred = my_log_pred[8459:8464],
           lda_pred = my_lda_pred[8459:8464],
           true = Default_train$default[8459:8464])
```

```
##   log_pred lda_pred true
## 1     <NA>     <NA> <NA>
## 2     <NA>     <NA> <NA>
## 3     <NA>     <NA> <NA>
## 4     <NA>     <NA> <NA>
## 5     <NA>     <NA> <NA>
## 6     <NA>     <NA> <NA>
```

```
## ----cm1, eval = FALSE---------------------------------------------------
## conf_lda <- table(my_lda_pred, Default$default)
```

```
## conf_lda


## ----ref.label = "cm1", echo = FALSE----------------------------------------
conf_lda <- table(my_lda_pred, Default_test$default)
conf_lda
```

```
##
## my_lda_pred   No  Yes
##         No  4783  146
##         Yes   70    1
```

```
conf_log <- table(my_log_pred, Default_test$default)
conf_log
```

```
##
## my_log_pred   No  Yes
##         No  4768  145
##         Yes   85    2
```

```
(1/nrow(Default_test)) * (conf_lda[2, 1] + conf_lda[1, 2])
```

```
## [1] 0.0432
```

```
(1/nrow(Default_test)) * (conf_log[2, 1] + conf_log[1, 2])
```

```
## [1] 0.046
```

```
#Calculate LDA misclssification rate
lda_mis = (70 + 146)/(4783 + 70 + 146 + 1)
log_mis = (85 + 145)/(4768 + 145 + 85 + 2)
lda_mis
```

```
## [1] 0.0432
```

```
log_mis
```

```
## [1] 0.046
```

The testing misclassification rate of logistic regression (0.046) is very slightly higher than the testing misclassification rate of LDA (0.0432).