Regressão Logística e medidas de performance

Tiago Mendonça dos Santos

tiagoms.com
tiagomendonca
tiagoms1@insper.edu.br

Regressão Logística

Classificação

O banco de dados *Default*[1] contém informações sobre 10.000 clientes. O objetivo nesse contexto é predizer quais clientes apresentarão *default* no cartão de crédito. A seguir são apresentadas as variáveis contidas no banco:

- **default**: indica se o cliente apresentou *default*
- **student**: indica se o cliente é estudante
- **balance**: saldo médio que o cliente tem sobrando no cartão de crédito após fazer o pagamento mensal
- **income**: renda do cliente

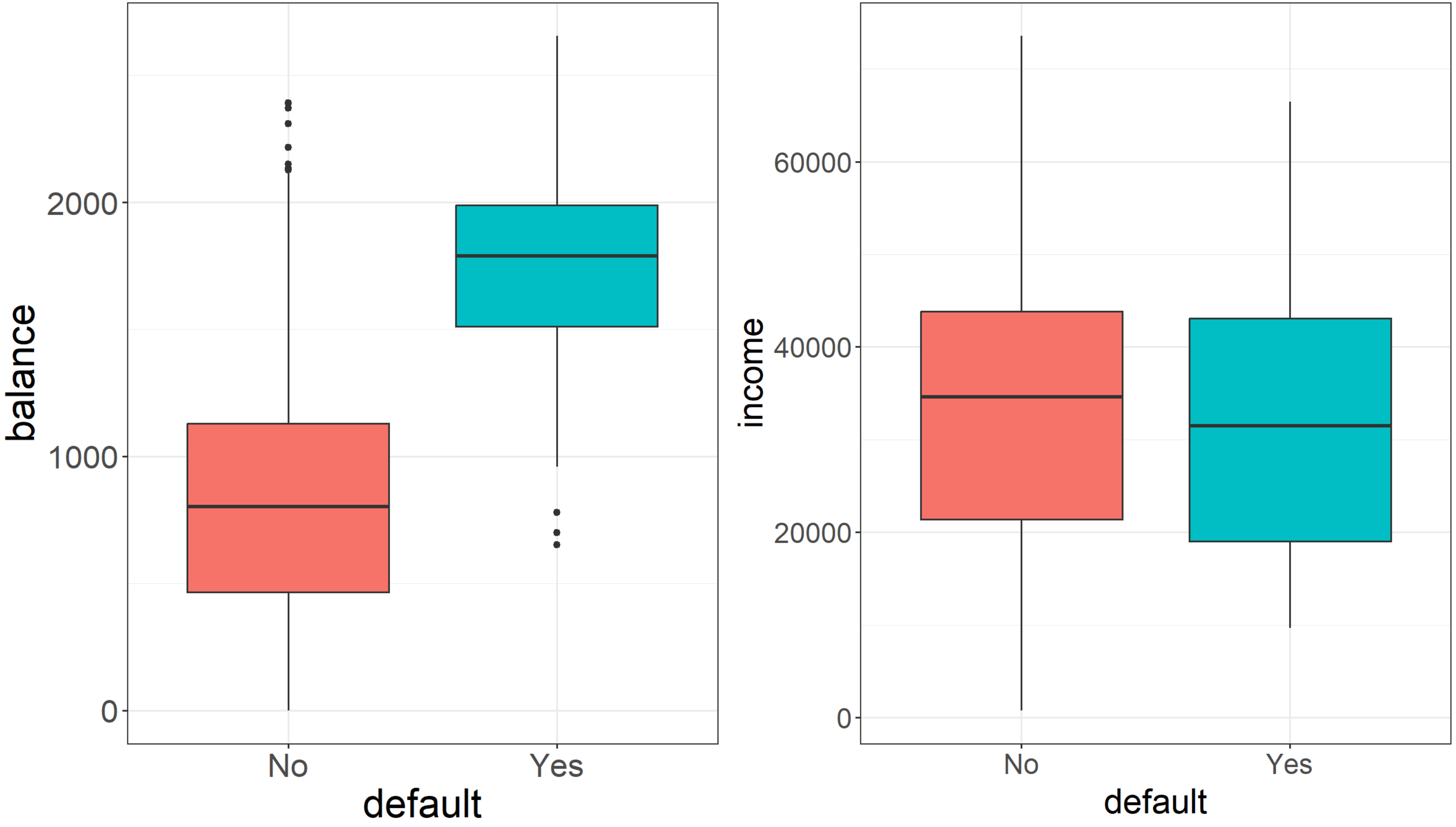
```r
library(ISLR)

str(Default)
```

```
## 'data.frame':	10000 obs. of  4 variables:
##  $ default: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ student: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ balance: num  730 817 1074 529 786 ...
##  $ income : num  44362 12106 31767 35704 38463 ...
```

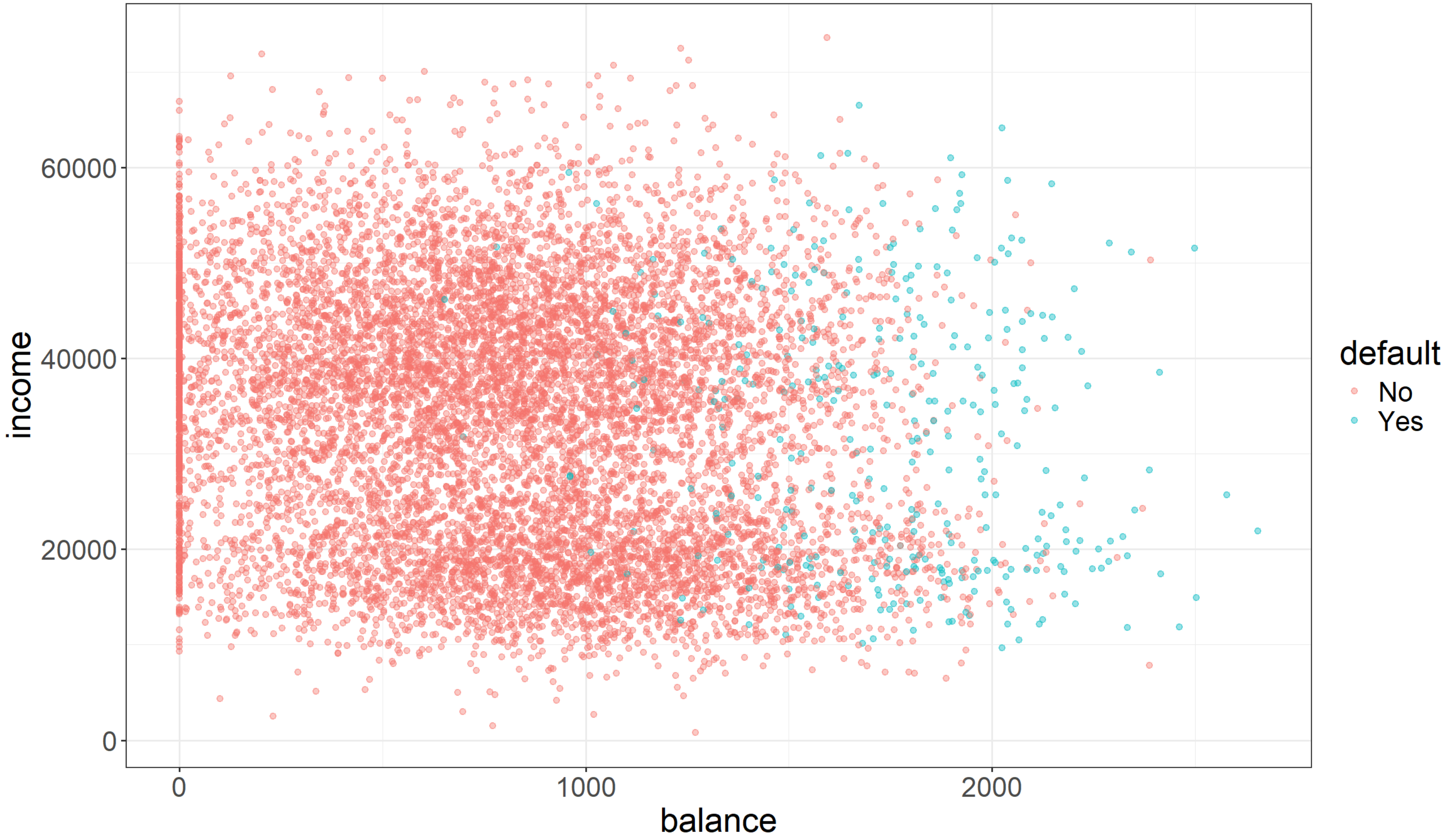
```r
# Default %>%
#   group_by(default) %>%
#   skim()
```

[1] os dados estão no pacote *ISLR*.

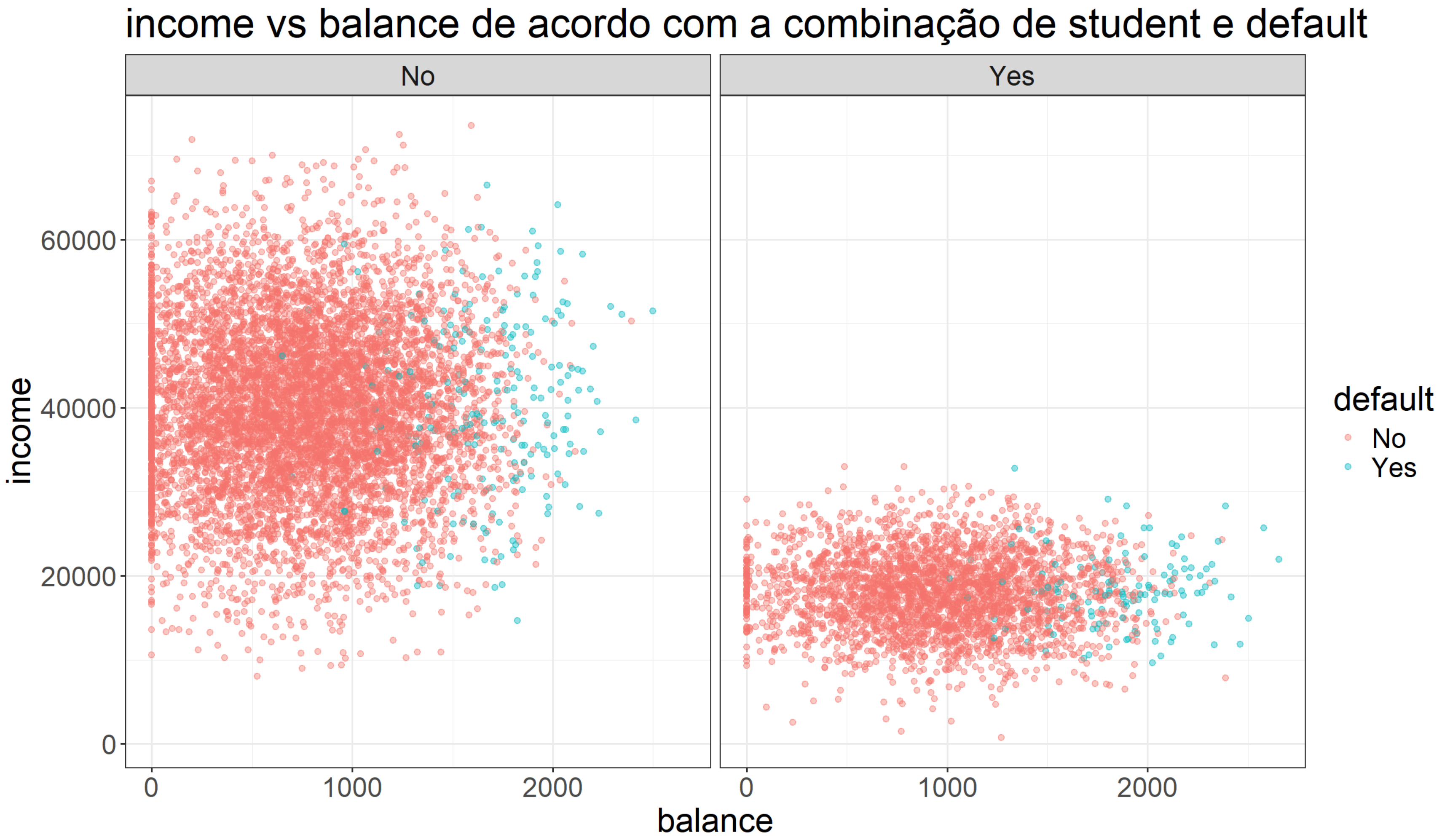
Classificação



Classificação



Classificação



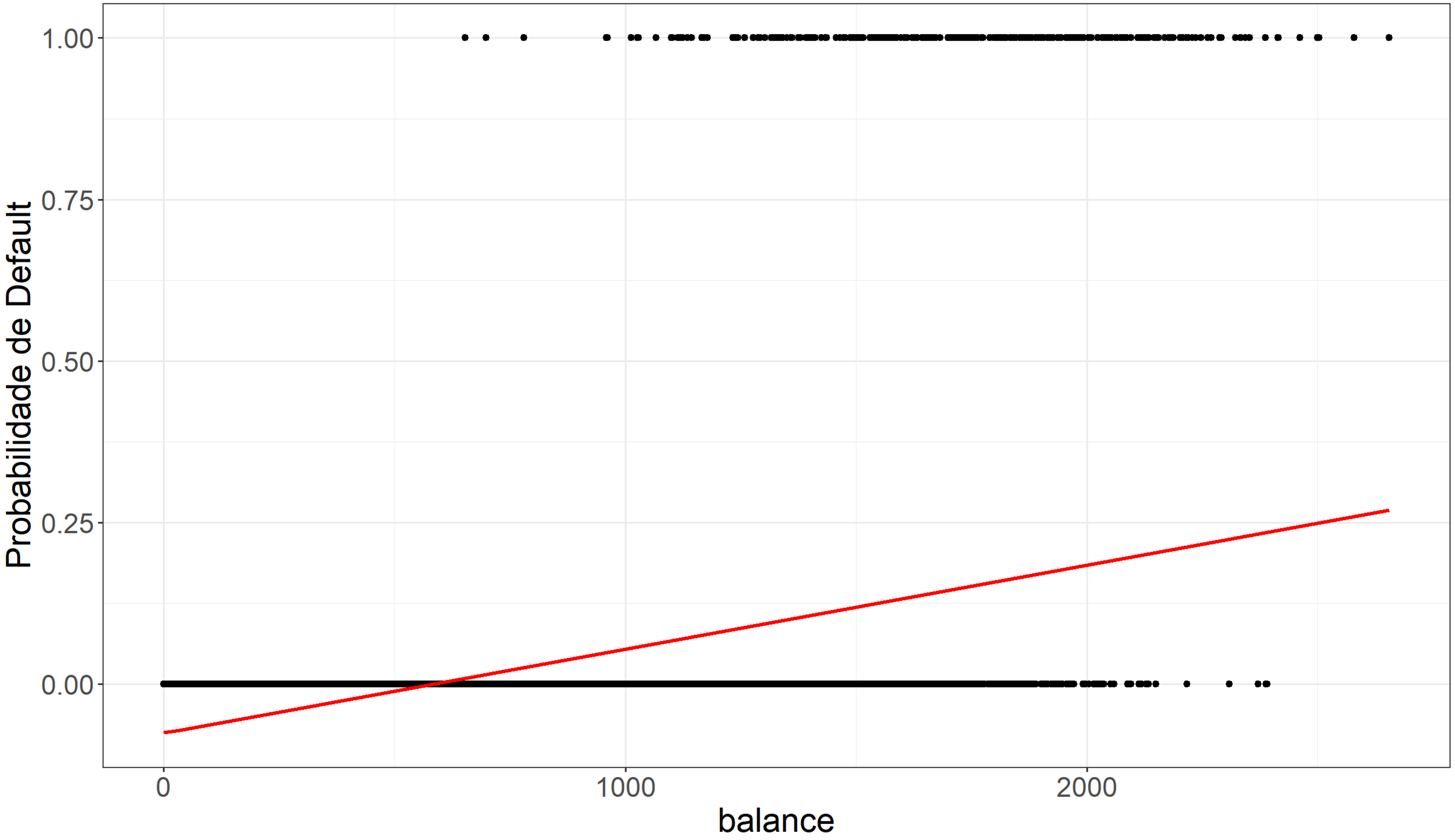
Classificação

Vamos considerar $Y = 1$ para *default* e $Y = 0$ para não *default* e modelar $p(X) = P(Y = 1|X)$ por

$$p(X) = \beta_0 + \beta_1 X,$$

em que X é o *balance*.

Classificação



Problema?

Insper

Alternativa - Modelar alguma função da chance

$$\text{chance} = \frac{p(X)}{1 - p(X)}$$

Probabilidade (0 e 1)	**Chance (0 e ∞)**
0.90	90:10 ou 9
0.75	75:25 ou 3
0.50	50:50 ou 1
0.20	20:80 ou 0.25
0.10	10:90 ou 0.1111
0.01	1:99 ou 0.0101

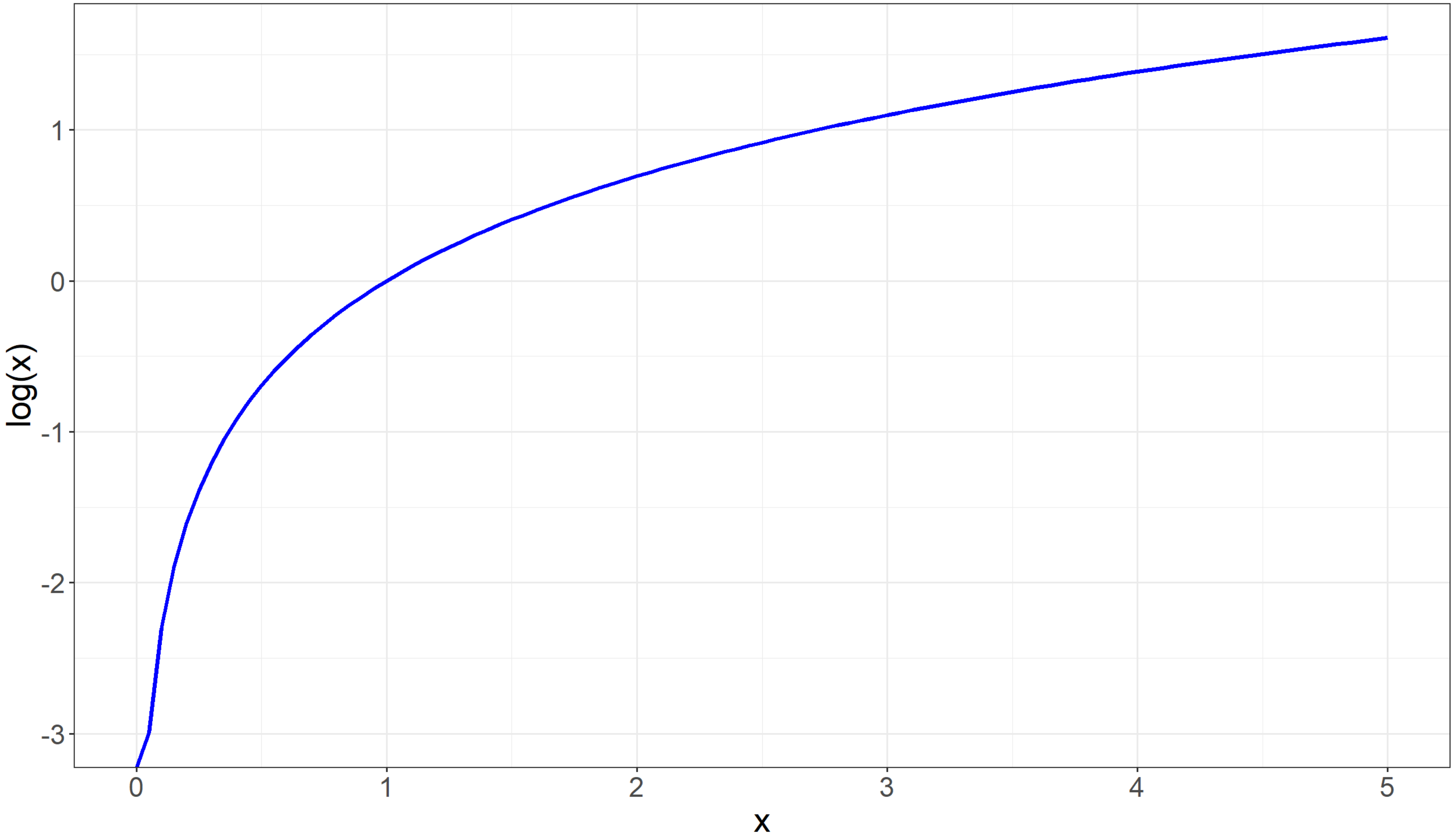
Insper

Alternativa - Modelar alguma função da chance

$$\text{chance} = \frac{p(X)}{1 - p(X)}$$

Probabilidade (0 e 1)	**Chance (0 e ∞)**	**Log da chance ($-\infty$ e $+\infty$)**
0.90	90:10 ou 9	2.197
0.75	75:25 ou 3	1.099
0.50	50:50 ou 1	0
0.20	20:80 ou 0.25	-1.386
0.10	10:90 ou 0.1111	-2.197
0.01	1:99 ou 0.0101	-4.595

Alternativa - Modelar alguma função da chance



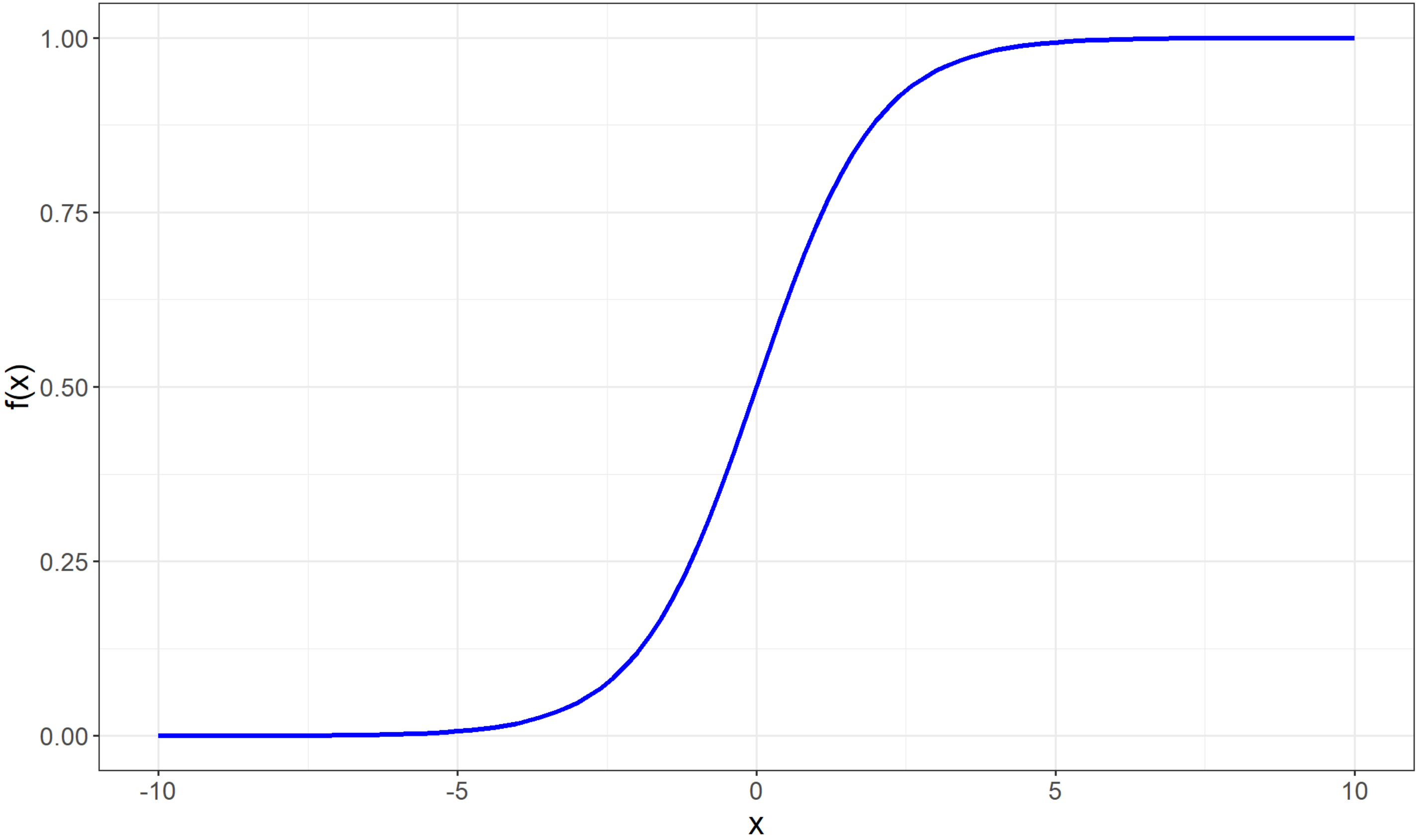
Alternativa - Modelar alguma função da chance

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

$$p(X) = \frac{\exp\{\beta_0 + \beta_1 X\}}{1 + \exp\{\beta_0 + \beta_1 X\}} = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 X)\}}$$

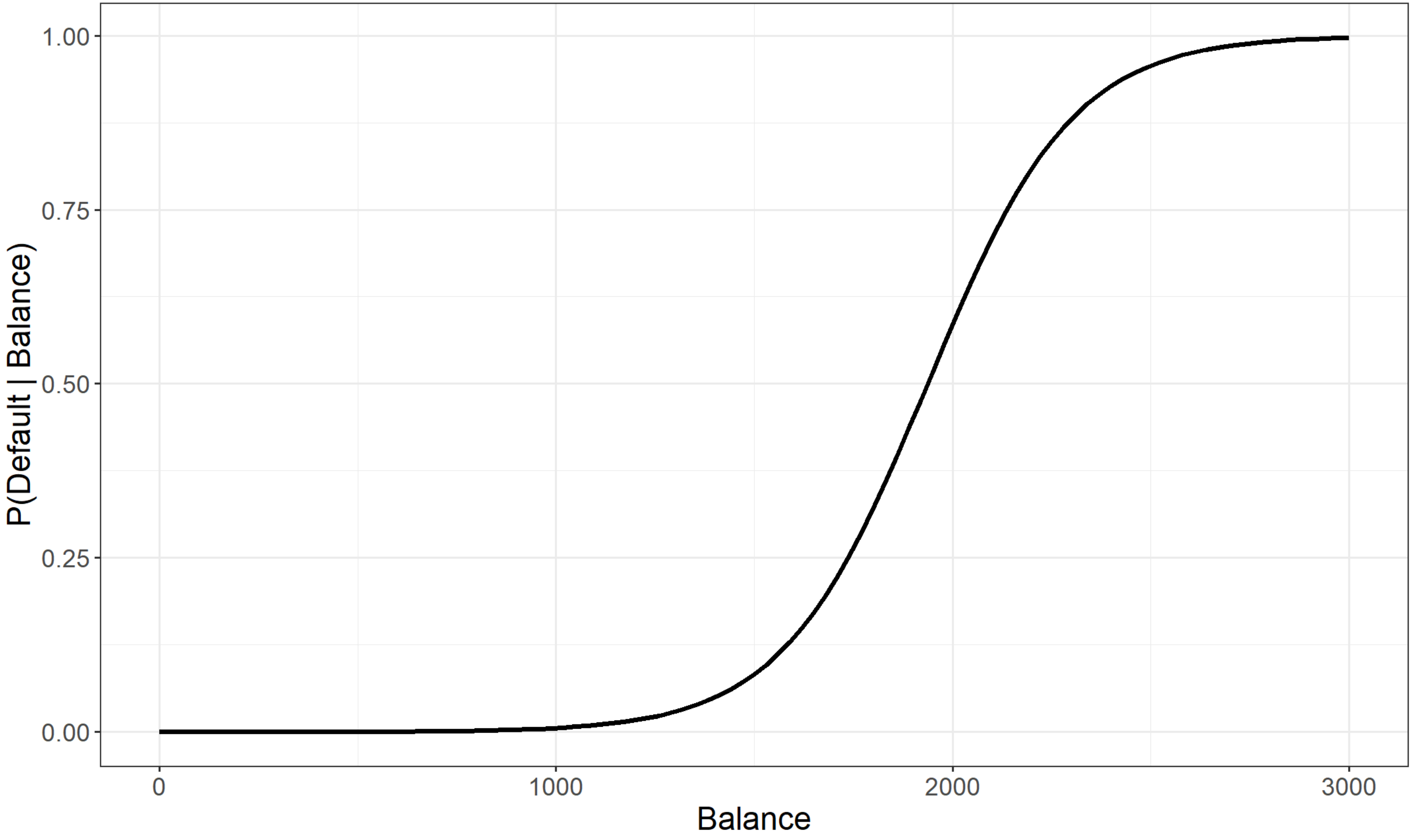
Logística

$$f(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$



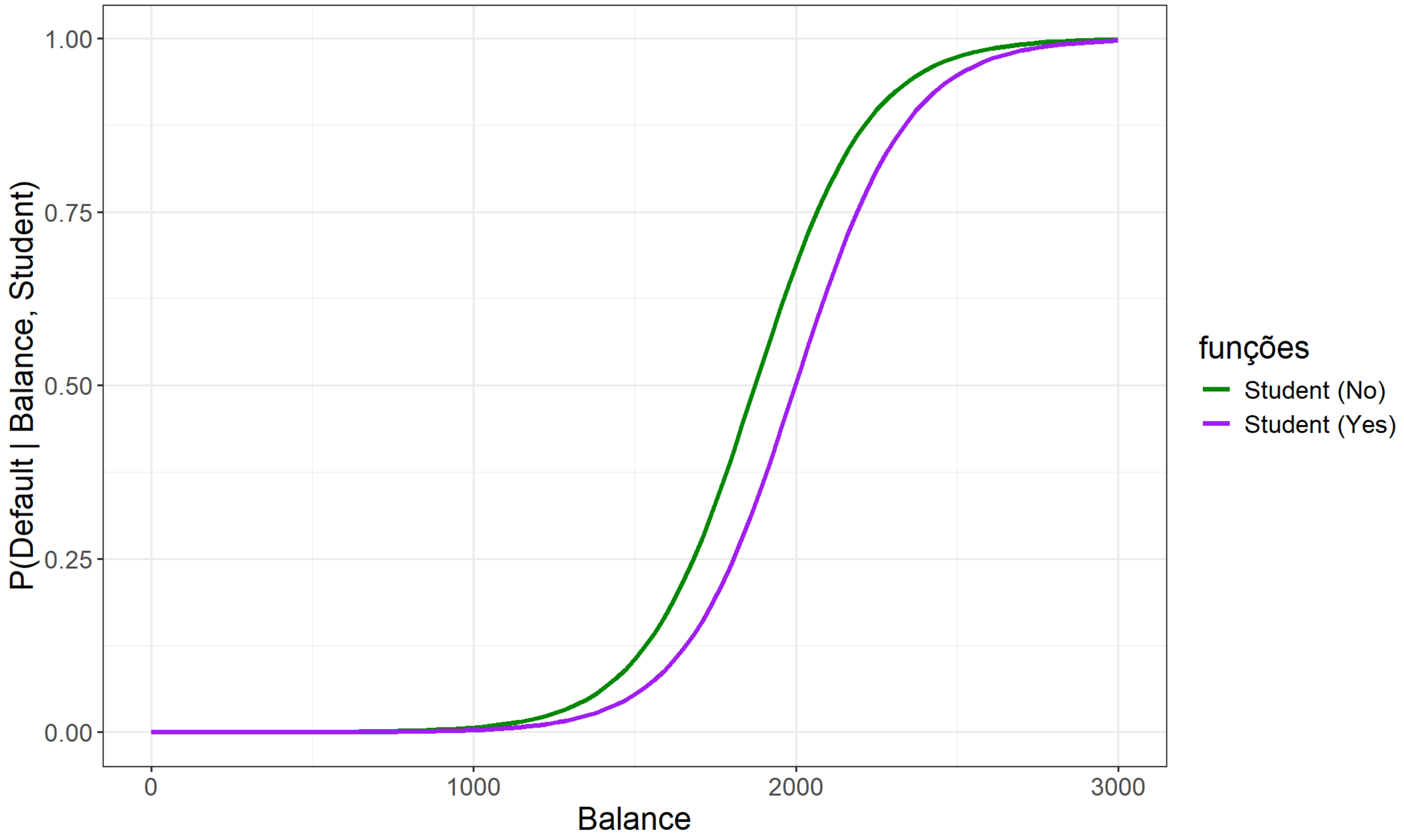
Logística

$$\log\left(\frac{p(X)}{1-p(X)}\right) = -10.6513 + 0.0055X$$



Logística

$$\log\left(\frac{p(X)}{1-p(X)}\right) = -10.7495 + 0.0057 \times Balance - 0.7145 \times Student$$



Como obter as estimativas para β

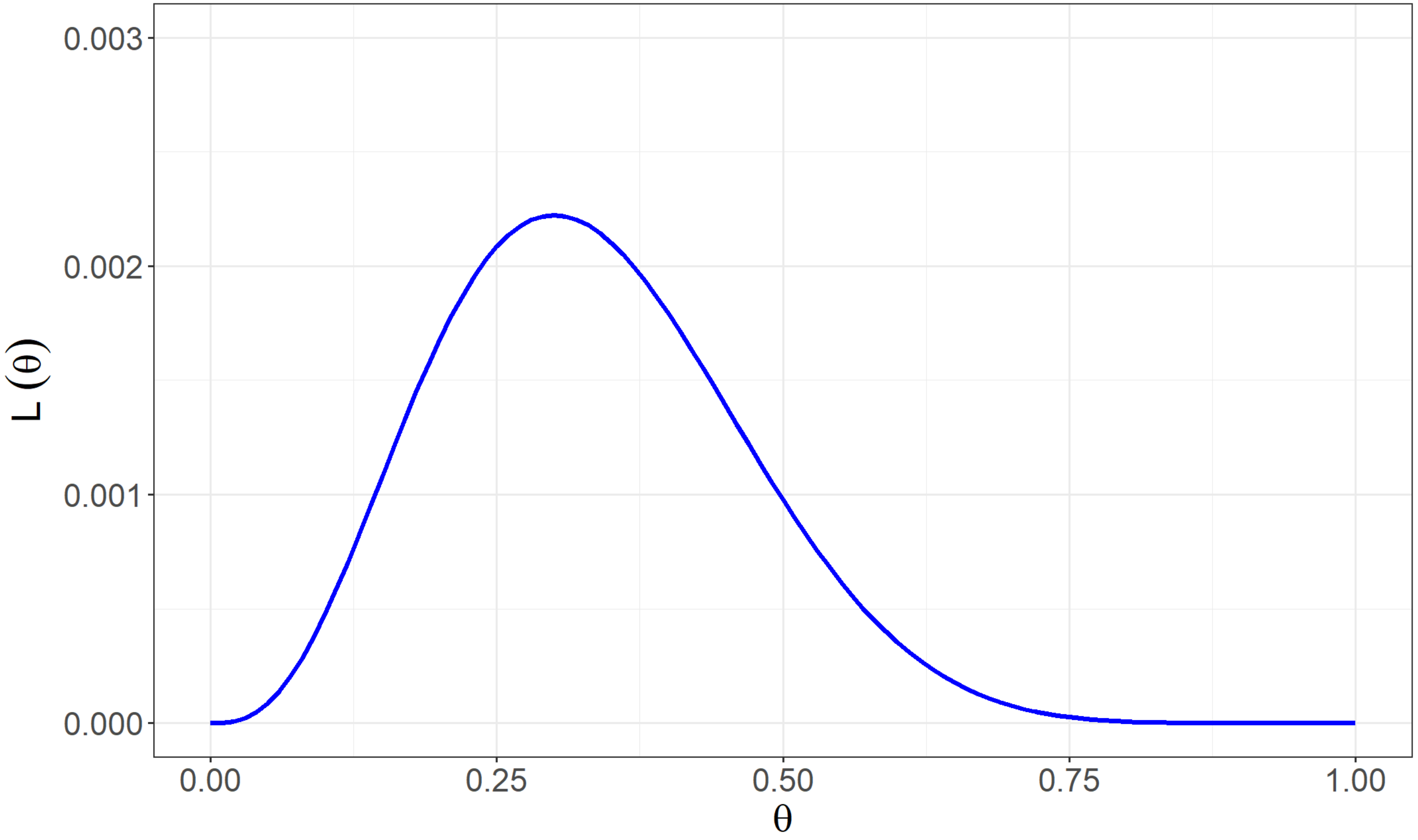
Vamos considerar uma situação mais simples em que todos apresentam a mesma probabilidade θ de apresentar *default* e foram observados d *defaults* numa amostra de tamanho n. Assim,

$$
\begin{aligned}
L_y(\theta) &= P(Y_1 = y_1, \ldots, Y_n = y_n|\theta) \\
&= \prod_{i=1}^{n} P(Y_i = y_i|\theta) \\
&= \prod_{i=1}^{n} \theta^{y_i}(1-\theta)^{1-y_i} \\
&= \theta^d(1-\theta)^{n-d}
\end{aligned}
$$

Exemplo

Tome $n = 10$ e $d = 3$.

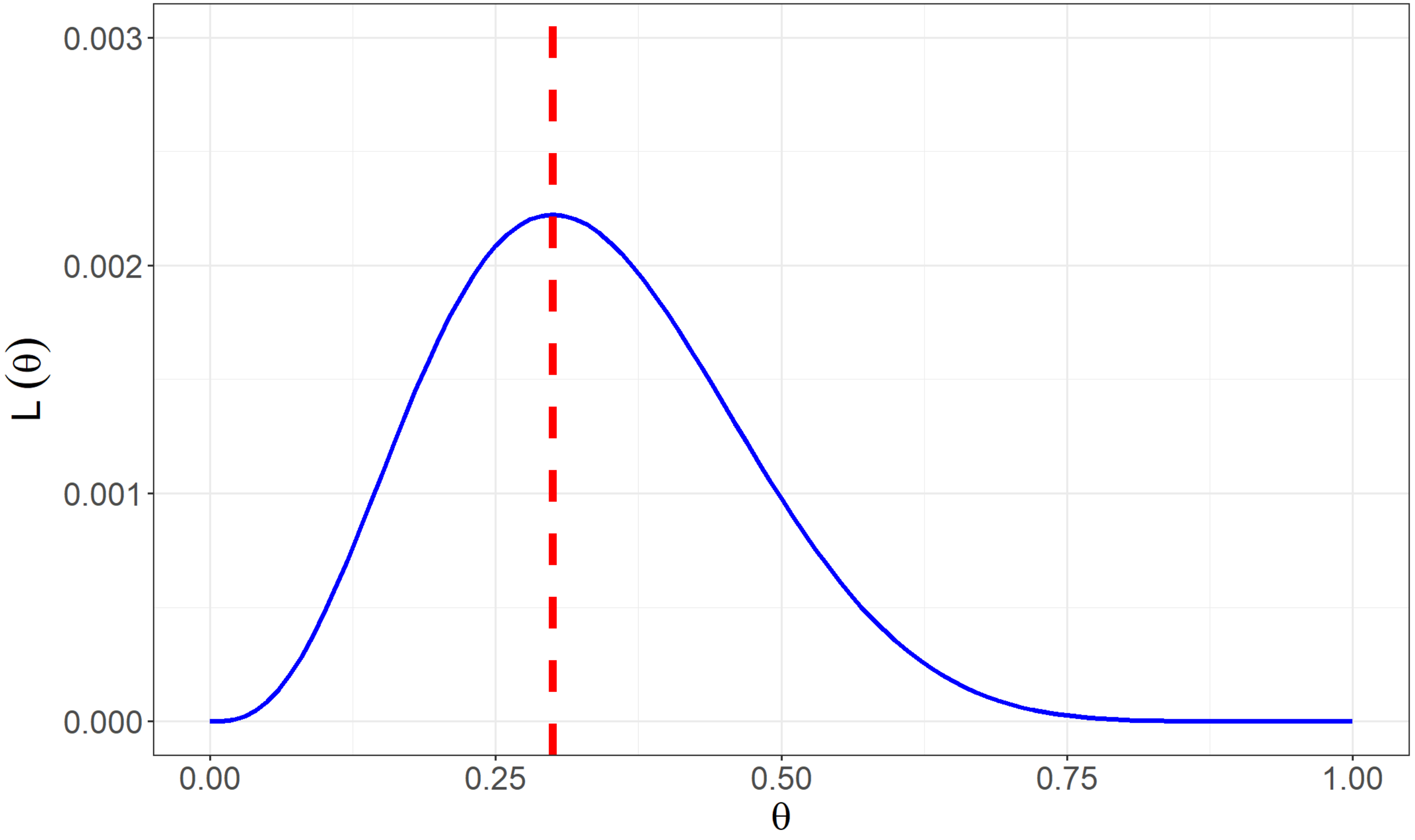
$$L_y(\theta) = \theta^3(1-\theta)^{10-3}$$



Exemplo

Tome $n = 10$ e $d = 3$.

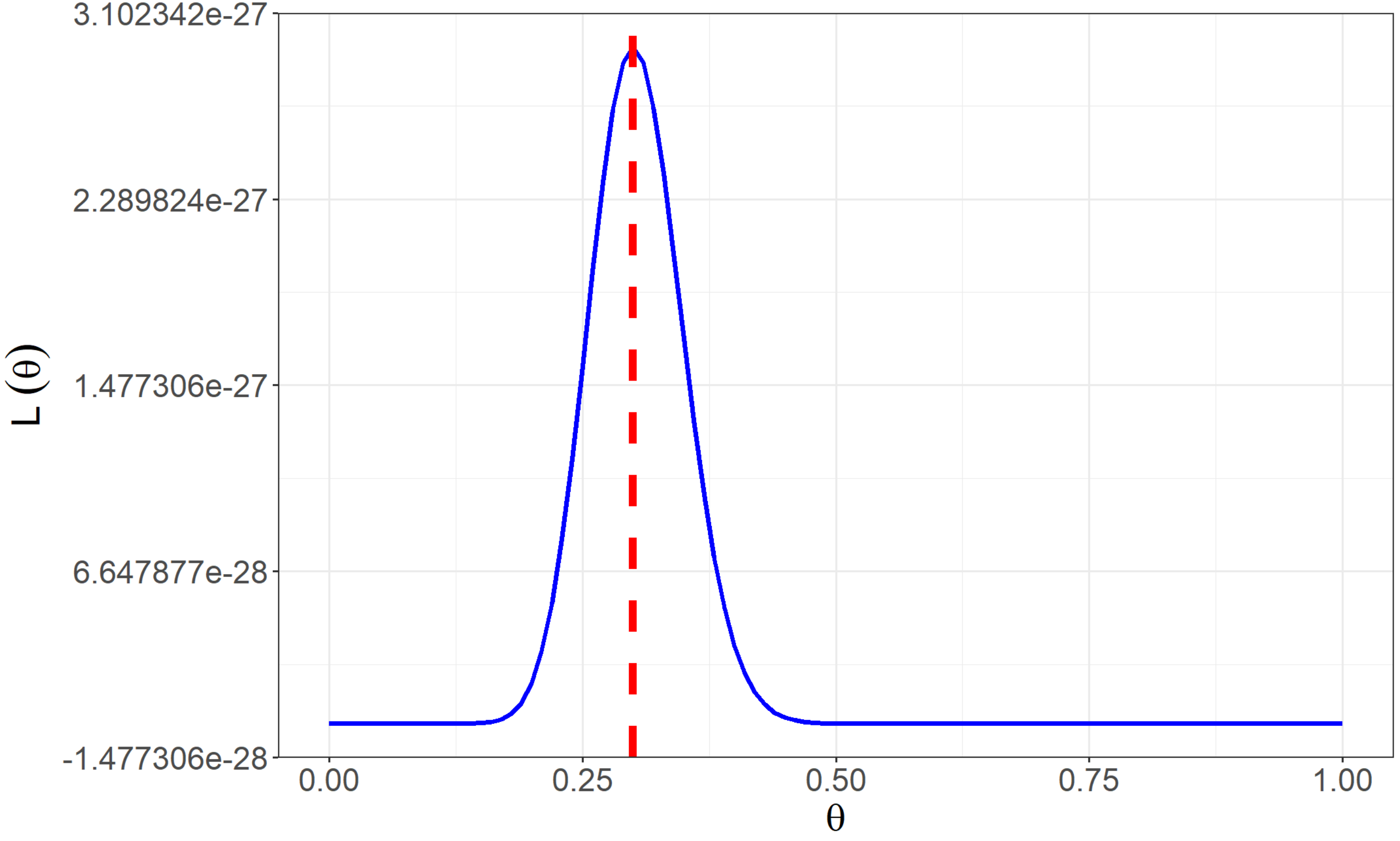
$$L_y(\theta) = \theta^3(1-\theta)^{10-3}$$



Exemplo

Tome $n = 100$ e $d = 30$.

$$L_y(\theta) = \theta^{30}(1-\theta)^{100-30}$$



Insper

Como obter as estimativas para β

Com a função de verossimilhança

$$
\begin{aligned}
L_{\mathbf{x},y}(\boldsymbol{\beta}) &= P(Y_1 = y_1, \ldots, Y_n = y_n | \mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\beta}) \\
&= \prod_{i=1}^{n} P(Y_i = y_i | \mathbf{x}_i, \boldsymbol{\beta}) \\
&= \prod_{i=1}^{n} p(\mathbf{x}_i)^{y_i} [1 - p(\mathbf{x}_i)]^{1-y_i} \\
&= \prod_{i=1}^{n} \left(\frac{\exp\{\beta_0 1 + \cdots + \beta_p x_{p,i}\}}{1 + \exp\{\beta_0 1 + \cdots + \beta_p x_{p,i}\}} \right)^{y_i} \left(1 - \frac{\exp\{\beta_0 1 + \cdots + \beta_p x_{p,i}\}}{1 + \exp\{\beta_0 1 + \cdots + \beta_p x_{p,i}\}} \right)^{1-y_i}
\end{aligned}
$$

Modelo

```r
fit <- glm(default ~ balance + student, data = Default, family = "binomial")

summary(fit)
```

```
##
## Call:
## glm(formula = default ~ balance + student, family = "binomial",
##     data = Default)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.075e+01  3.692e-01 -29.116  < 2e-16 ***
## balance      5.738e-03  2.318e-04  24.750  < 2e-16 ***
## studentYes  -7.149e-01  1.475e-01  -4.846 1.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.7  on 9997  degrees of freedom
## AIC: 1577.7
##
## Number of Fisher Scoring iterations: 8
```

Interpretação - Odds ratio - Razão de chances

Seja X_1 o perfil com balance = 1000 e estudante e X_0 o perfil com balance = 1000 e não estudante.

$$\text{odds ratio} = \frac{\frac{p(X_1)}{1-p(X_1)}}{\frac{p(X_0)}{1-p(X_0)}} = \frac{\exp\{\beta_0 + \beta_1 \times 1000 + \beta_2 \times 1\}}{\exp\{\beta_0 + \beta_1 \times 1000 + \beta_2 \times 0\}} = \exp\{\beta_2\}$$

Portanto, o perfil de estudante está associado a uma redução de 51% (exp(-0.714)) na chance de default em relação à chance do perfil não estudante.

Seja X_1 o perfil com balance = 1100 e estudante e X_0 o perfil com balance = 1000 e estudante.

$$\text{odds ratio} = \frac{\frac{p(X_1)}{1-p(X_1)}}{\frac{p(X_0)}{1-p(X_0)}} = \frac{\exp\{\beta_0 + \beta_1 \times 1100 + \beta_2 \times 1\}}{\exp\{\beta_0 + \beta_1 \times 1000 + \beta_2 \times 1\}} = \exp\{\beta_1 \times 100\}$$

Portanto, o aumento de 100 unidade no balance está associado a um aumento de 77% (exp(0.0057*100)) na chance de default em relação à indivíduos com 100 unidade a menos de balance.

Como obter as estimativas para β

Com a função de verossimilhança

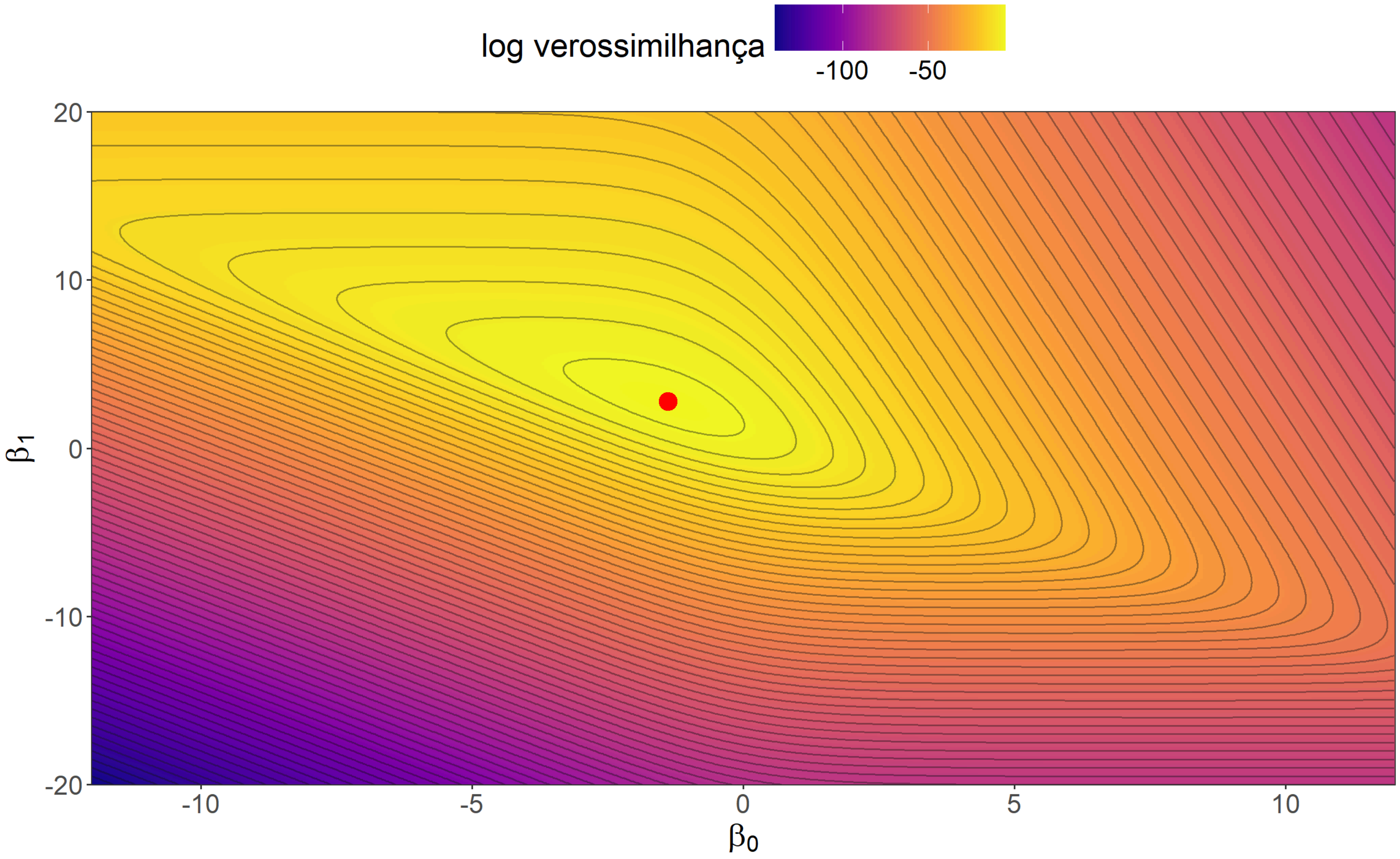
$$
\begin{aligned}
L_{\mathbf{x},y}(\boldsymbol{\beta}) &= P(Y_1 = y_1, \ldots, Y_n = y_n | \mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\beta}) \\
&= \prod_{i=1}^{n} P(Y_i = y_i | \mathbf{x}_i, \boldsymbol{\beta}) \\
&= \prod_{i=1}^{n} p(\mathbf{x}_i)^{y_i} [1 - p(\mathbf{x}_i)]^{1-y_i} \\
&= \prod_{i=1}^{n} \left(\frac{\exp\{\beta_0 + \beta_1 x_{1,i}\}}{1 + \exp\{\beta_0 + \beta_1 x_{1,i}\}} \right)^{y_i} \left(1 - \frac{\exp\{\beta_0 + \beta_1 x_{1,i}\}}{1 + \exp\{\beta_0 + \beta_1 x_{1,i}\}} \right)^{1-y_i}
\end{aligned}
$$

Exemplo

```r
dados <- tribble(~x,  ~y,
                  0,   0,
                  0,   0,
                  0,   0,
                  0,   0,
                  0,   1,
                  1,   0,
                  1,   1,
                  1,   1,
                  1,   1,
                  1,   1)

coef(summary(glm(y ~ x, family = "binomial", data = dados)))
```

```
##              Estimate Std. Error  z value   Pr(>|z|)
## (Intercept) -1.386294   1.118034 -1.23994 0.21499771
## x            2.772589   1.581138  1.75354 0.07950944
```

Exemplo



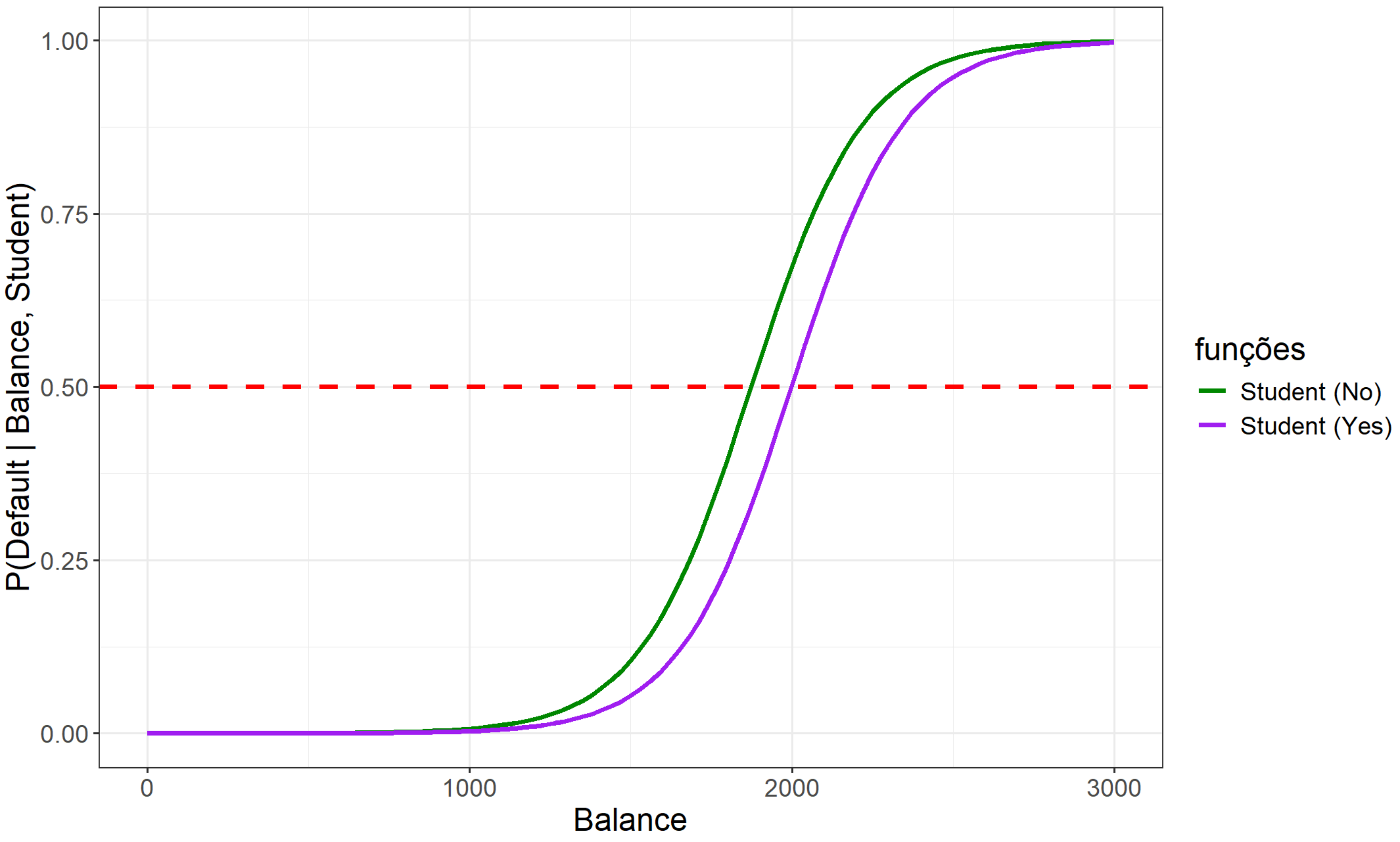
Exemplo

Para maximizar a função de log verossimilhança, podemos utilizar um otimizador qualquer. Nesse exemplo, considero a função `optim`. Note que o padrão desse otimizador é minimizar a função. No nosso caso, para maximizar, podemos apenas trocar o sinal da função ou utilizar o argumento `control = list(fnscale = -1)`.

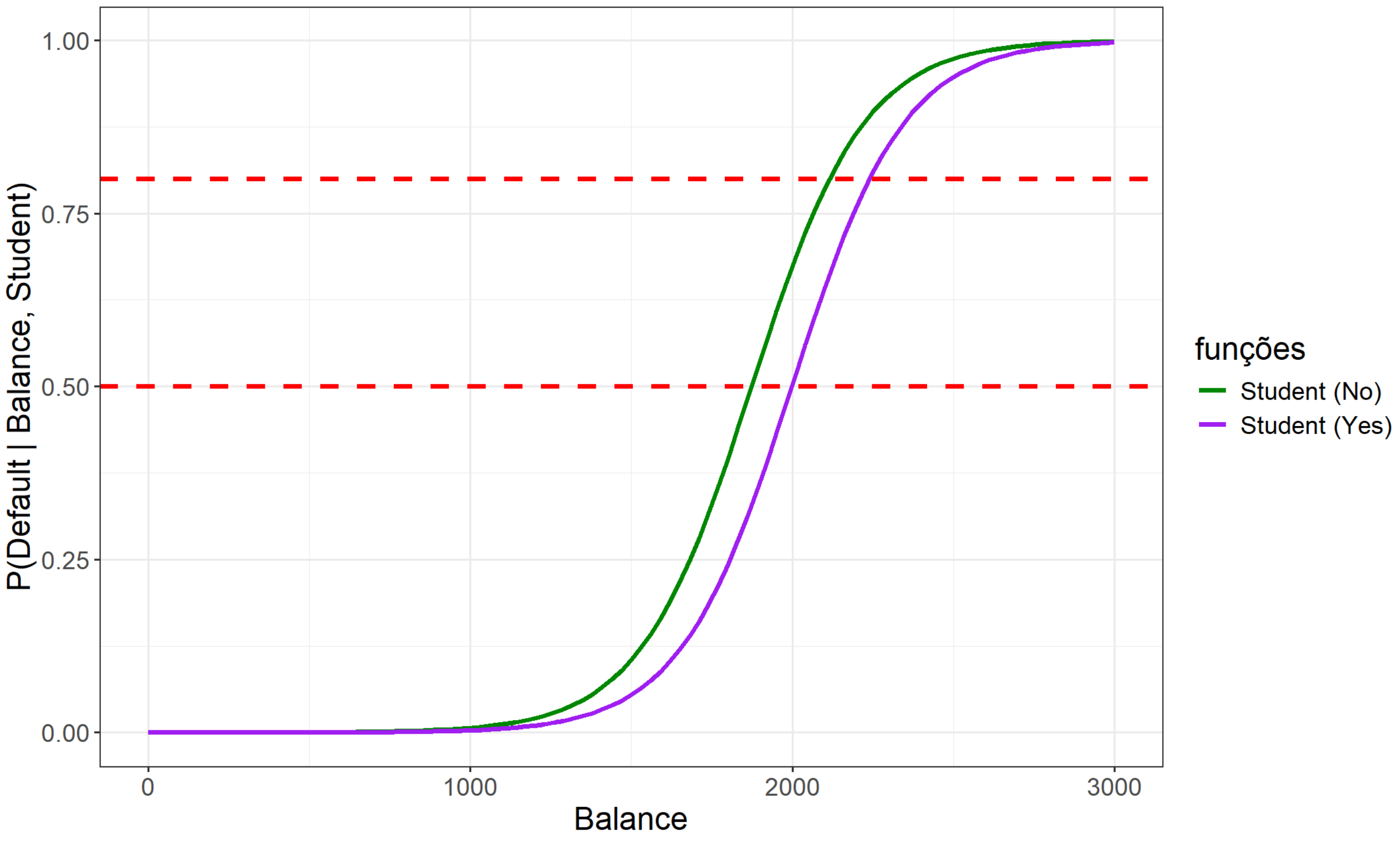
```
log_vero <- function(x) {

  b0 <- x[1]
  b1 <- x[2]

  sum(dados$y * log( exp(b0 + b1*dados$x) / (1 + exp(b0 + b1*dados$x))) +
        (1 - dados$y) * log((1 - ( exp(b0 + b1*dados$x) / (1 + exp(b0 + b1*dados$x))))))

}

optim(par = c(0, 0), log_vero, control = list(fnscale = -1))$par
```

```
## [1] -1.385860  2.771888
```

Como classificar?



Como classificar?



Treino e teste

Utilizaremos a abordagem treinameto/teste com 70%/30% e faremos a divisão proporcional a variável resposta.

```
library(rsample)

set.seed(123)
splits <- initial_split(Default, prop = .7, strata = default)

tr <- training(splits)
test <- testing(splits)

tr %>% count(default)
```

```
##   default    n
## 1      No 6770
## 2     Yes  230
```

```
test %>% count(default)
```

```
##   default    n
## 1      No 2897
## 2     Yes  103
```

```
fit <- glm(default ~ ., data = tr, family = "binomial")
prob <- predict(fit, test, type = "response")
```

Treino e teste

Para **corte = 0.2**

```
##                observado
## classificacao   No  Yes
##            No 2821   43
##           Yes   76   60
```

Para **corte = 0.5**

```
##                observado
## classificacao   No  Yes
##            No 2889   74
##           Yes    8   29
```

Para **corte = 0.8**

```
##                observado
## classificacao   No  Yes
##            No 2897   94
##           Yes    0    9
```

Faça extração de métricas de acerto para o corte de 0.2. Interprete essas métricas.

Medidas de performance

Erro de classificação total: $\frac{b+c}{n} = 1 - \frac{a+d}{n}$

Verdadeiro Positivo (Sensibilidade ou Recall): $\frac{d}{b+d}$

Verdadeiro Negativo (Especificidade): $\frac{a}{a+c}$

Valor Preditivo Positivo (Precision): $\frac{d}{c+d}$

Valor Preditivo Negativo: $\frac{a}{a+b}$

F-score (mede a performance na classe positiva): $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

	Observado	
Classificação	**No**	**Yes**
No	a	b
Yes	c	d

Medidas

Insper

```r
(roc_log <- roc(test$default, prob))
```

```
##
## Call:
## roc.default(response = test$default, predictor = prob)
##
## Data: prob in 2897 controls (test$default No) < 103 cases (test$default Yes).
## Area under the curve: 0.9447
```

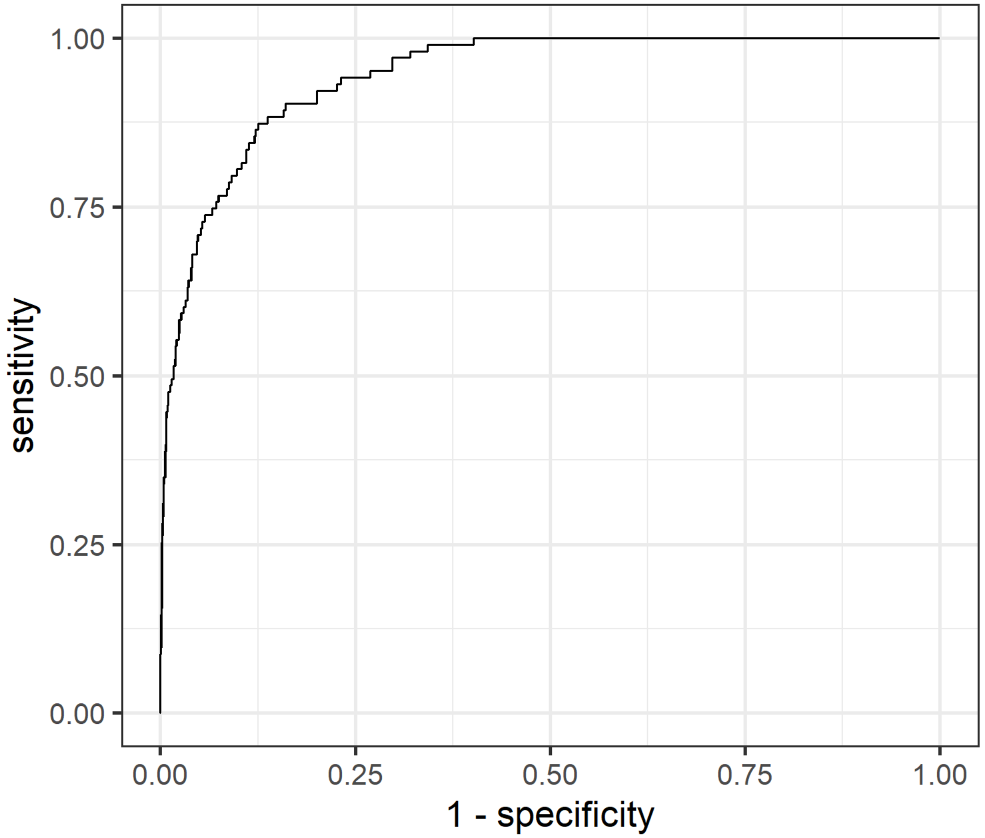
```r
coords(roc_log, c(.2, .5, .8),
       ret = c("threshold", "accuracy", "specificity", "sensitivity", "npv", "ppv"))
```

```
##   threshold  accuracy specificity sensitivity       npv       ppv
## 1       0.2 0.9603333   0.9737660  0.58252427 0.9849860 0.4411765
## 2       0.5 0.9726667   0.9972385  0.28155340 0.9750253 0.7837838
## 3       0.8 0.9686667   1.0000000  0.08737864 0.9685724 1.0000000
```

```r
# execute com ret = c("all")
```

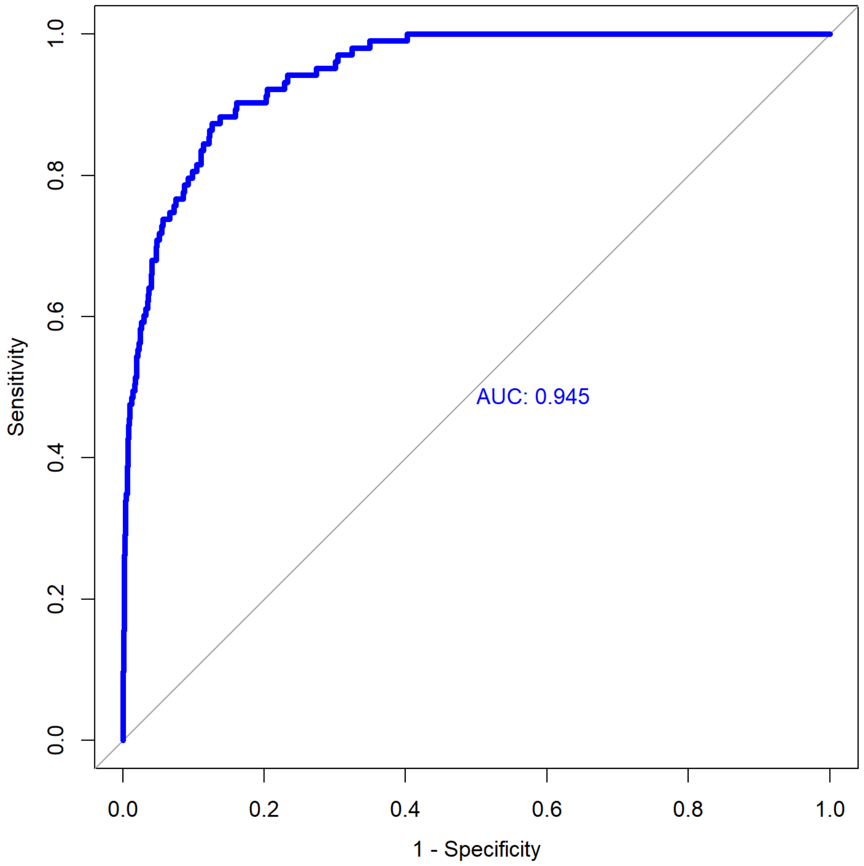
Curva ROC

```
coords(roc_log, seq(0, 1, 0.0005), ret = c("threshold", "specificity", "sensitivity")) %>%
  as_tibble() %>%
  ggplot(aes(1 - specificity, sensitivity)) +
    geom_step(direction = "vh")
```



Curva ROC

```r
plot(roc_log, legacy.axes = TRUE, print.auc = TRUE, lwd = 4, col = "blue")
```



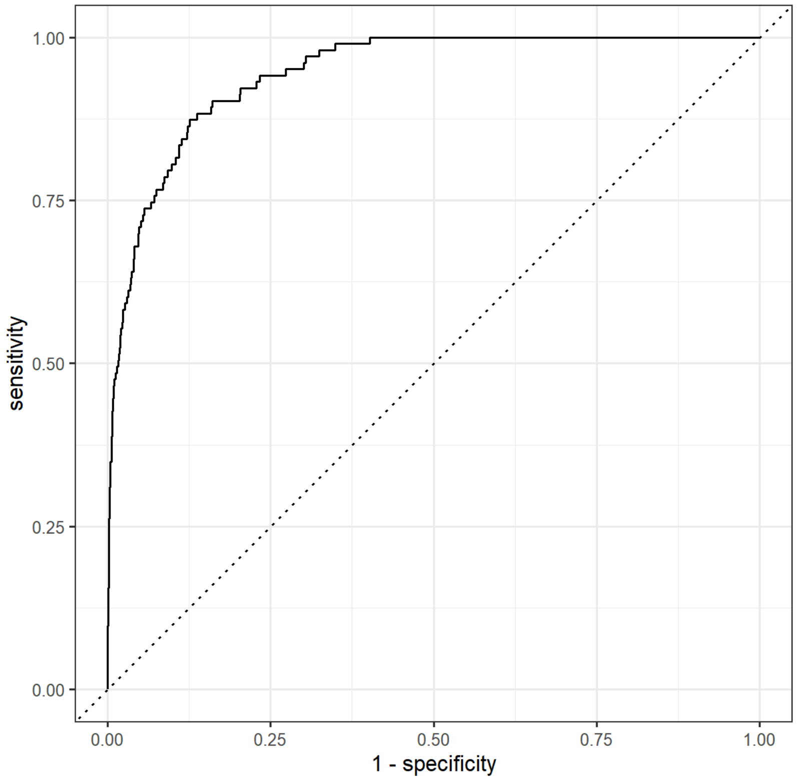
A curva ROC (Receiver Operating Characteristic) apresenta em uma única figura o desempenho de acordo do classificador de acordo com diferentes pontos de corte.

A área sob a curva (area under the curve - AUC) ROC pode ser interpretada como a probabilidade de uma observação do grupo positivo/churn/evento ter um marcador/probabilidade/propensão maior do que um indivíduo do grupo de referência, ou seja,

$$\mathrm{AUC} = P(\mathrm{M}_1 > \mathrm{M}_2).$$

Alternativa para curva ROC

```
library(yardstick)

tibble(classe = test$default, marcador = prob) %>%
  roc_curve(classe, marcador, event_level = "second") %>%
  autoplot()
```



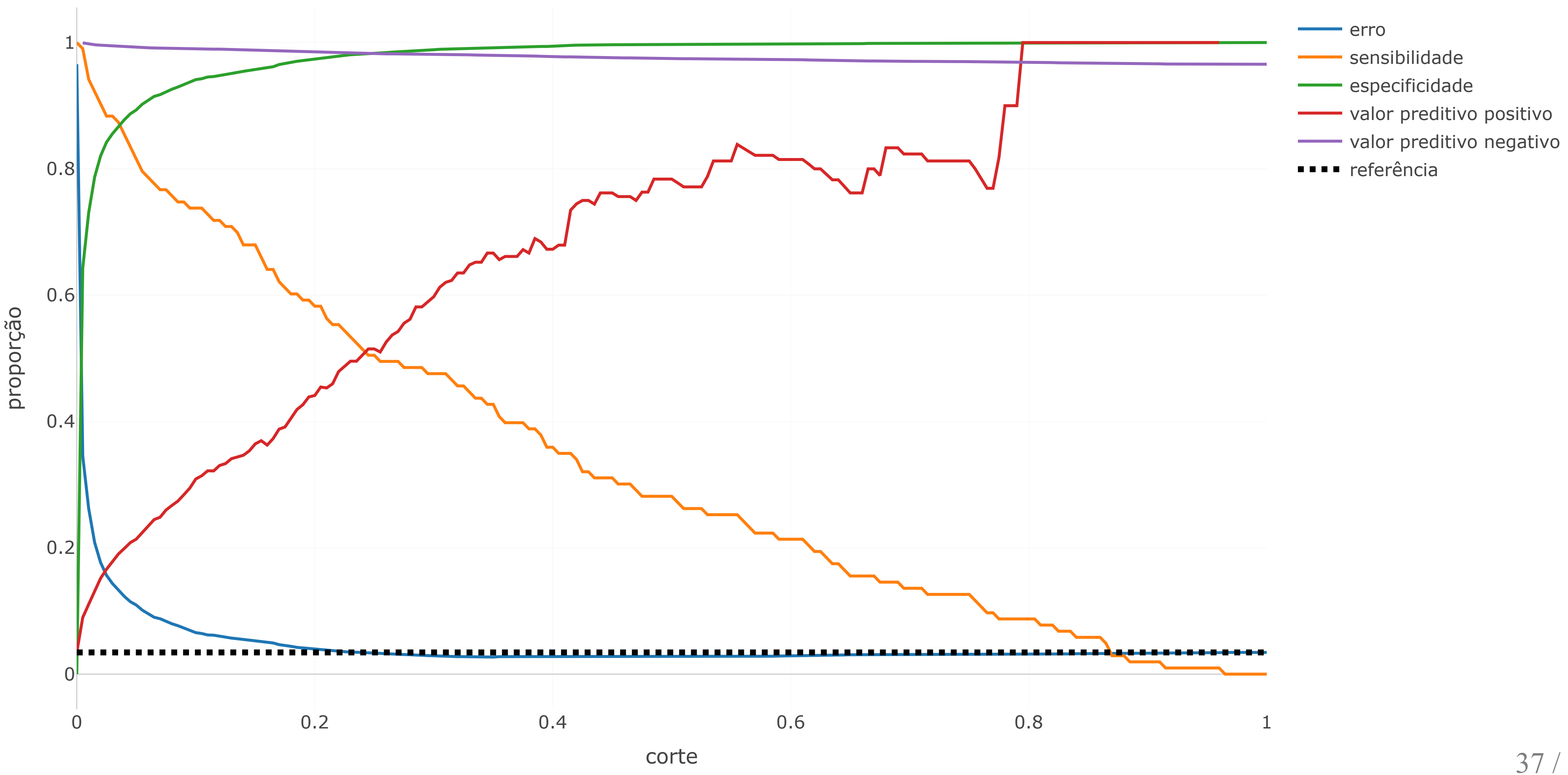
Alternativa para curva ROC

Insper

```r
library(yardstick)

tibble(classe = test$default, marcador = prob) %>%
  roc_auc(classe, marcador, event_level = "second")
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.945
```

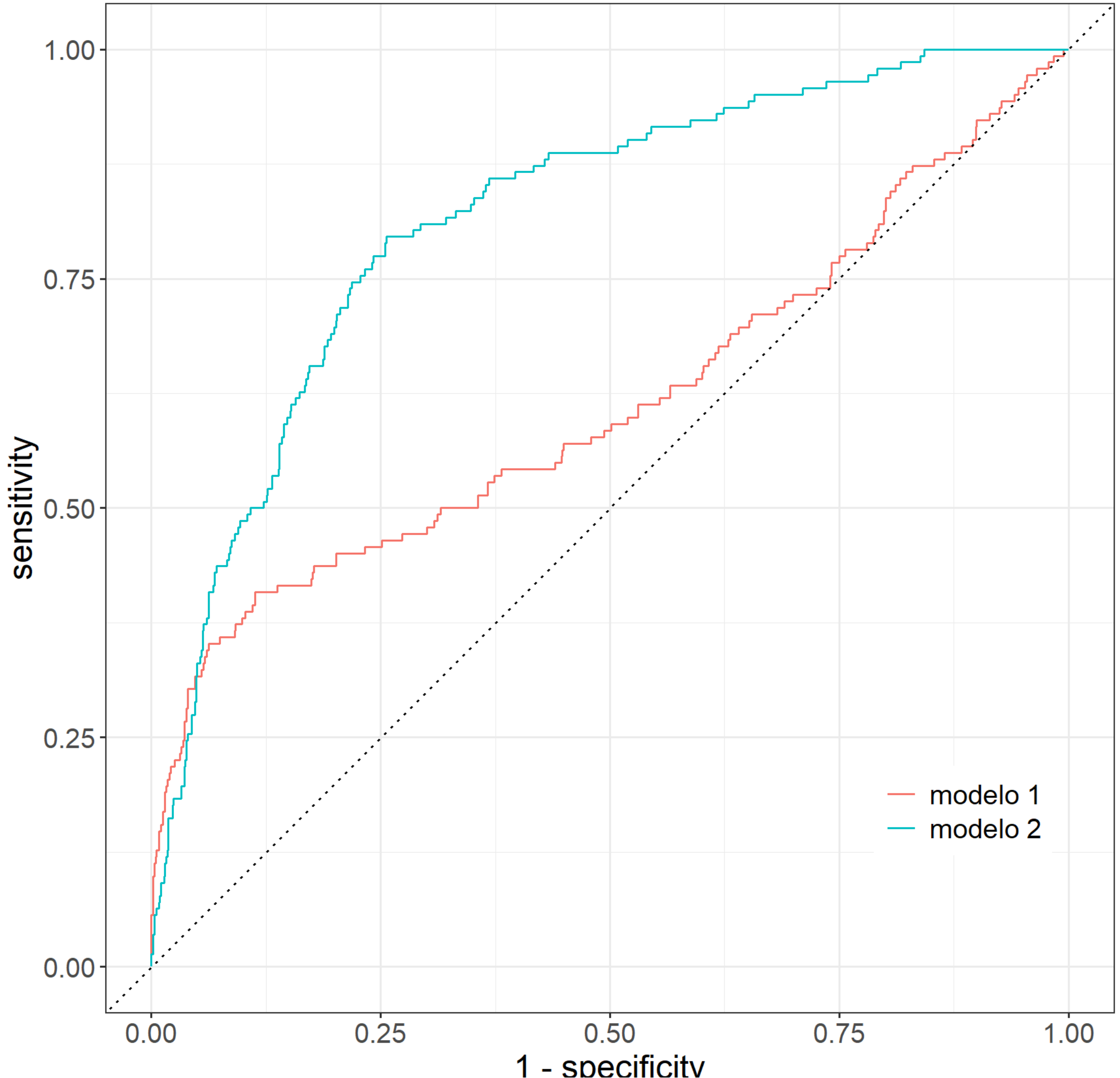
Medidas


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
proporção
corte
erro
sensibilidade
especificidade
valor preditivo positivo
valor preditivo negativo
referência

Exemplo - Churn

```r
library(modeldata)
library(rsample)
library(yardstick)

data(mlc_churn)

mlc_churn <- mlc_churn %>%
  mutate(churn = factor(churn, levels = c("no", "yes")))

# treinamento x teste -----------------------------------------------------

set.seed(313)

splits <- initial_split(mlc_churn, prop = .8, strata = "churn")

treinamento <- training(splits)
teste       <- testing(splits)

# logística 1 -------------------------------------------------------------

fit1 <- glm(churn ~ total_day_minutes + total_night_minutes,
            family = "binomial", data = treinamento)

# logística 2 -------------------------------------------------------------

fit2 <- glm(churn ~ ., family = "binomial", data = treinamento)
```

Avaliação


1.00
0.75
0.50
0.25
0.00
sensitivity
0.00
0.25
0.50
0.75
1.00
1 - specificity
modelo 1
modelo 2

Avaliação

Primeiro criamos um data frame com as probabilidades, classe e a identificação do modelo. As avaliações serão agrupadas de acordo com o modelo preditivo.

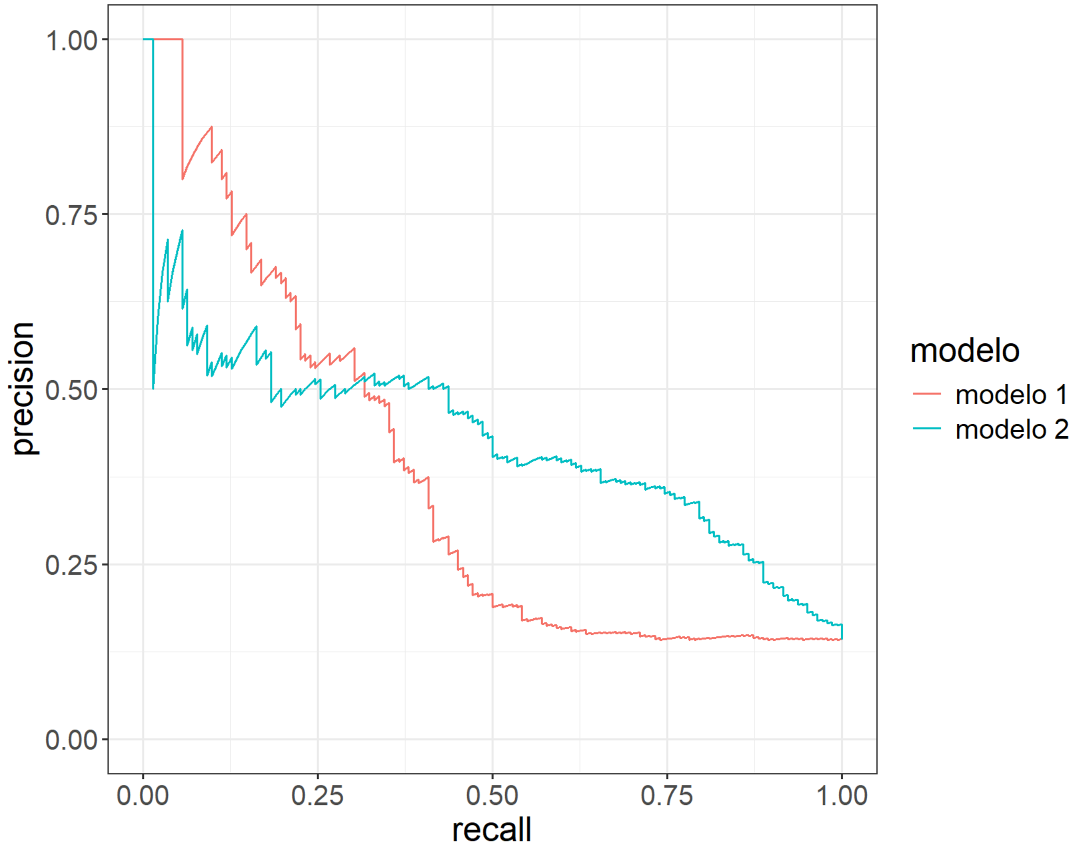
```
df_avaliacao <- tibble(probabilidade = predict(fit1, teste, type = "response"),
                       churn = teste$churn,
                       modelo = "modelo 1") %>%
  bind_rows(tibble(probabilidade = predict(fit2, teste, type = "response"),
                   churn = teste$churn,
                   modelo = "modelo 2"))
```

Primeiro criamos um data frame com as probabilidades, classe e a identificação do modelo. As avaliações serão agrupadas de acordo com o modelo preditivo.

```
df_avaliacao %>%
  group_by(modelo) %>%
  roc_curve(churn, probabilidade, event_level = "second") %>%
  autoplot() +
  labs(color = NULL) +
  theme(legend.position = c(.85, .2),
        text = element_text(size = 17))
```

Precision (VPP) x Recall (Sensibilidade)

Útil para encontrar um ponto de equilíbrio. Pensando em bloquear conta de usuários, queremos que a maioria das fraudes sejam identificadas (recall/sensibilidade) e garantir que os bloqueios de contas sejam feitos com acurácia.



Precision (VPP) x Recall (Sensibilidade)

Para gerar o gráfico, utilizamos

```
df_avaliacao %>%
  group_by(modelo) %>%
  pr_curve(churn, probabilidade, event_level = "second") %>%
  autoplot()
```

Para calcular a área sob a curva, podemos utilizar

```
df_avaliacao %>%
  group_by(modelo) %>%
  pr_auc(churn, probabilidade, event_level = "second")
```

modelo	.metric	.estimator	.estimate
modelo 1	pr_auc	binary	0.3688540
modelo 2	pr_auc	binary	0.4275696

Detecção de Fraudes em Transações Financeiras

Insper

Contexto: empresa financeira deseja detectar fraudes em transações com cartão de crédito.

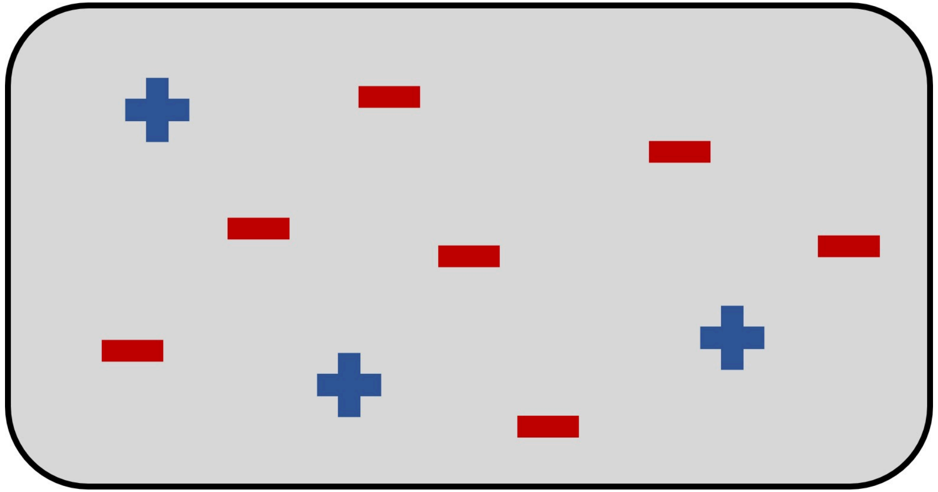
Objetivo: identificar transações fraudulentas para bloqueá-las e proteger os clientes, sem bloquear transações legítimas.

Curva de Ganho: utilizada para avaliar o desempenho do modelo de detecção de fraudes. Mostra o percentual de transações fraudulentas identificadas ao analisar uma certa porcentagem de transações.

Exemplo: ao analisar 20% das transações (maior probabilidade de fraude), identificamos 80% das transações fraudulentas. Com essa porcentagem de casos menor, podemos bloquear ou revisar manualmente as transações, reduzindo consideravelmente o impacto das fraudes. A curva de ganho mostra a diferença de desempenho ao utilizar o modelo em questão em relação a uma análise aleatória dos dados.

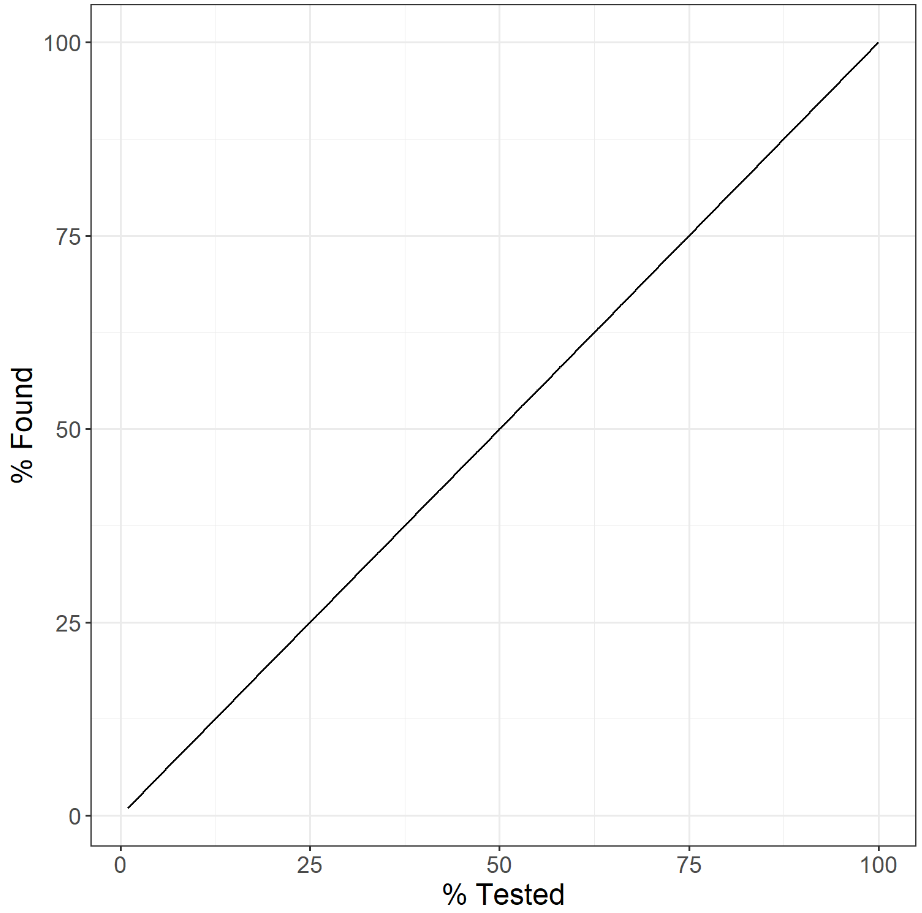
Modelo nulo ou aleatório

Se selecionarmos aleatoriamente 1/3 dos dados, esperamos coletar 1/3 dos casos e 1/3 dos controles.



Portanto, coletando 10% dos dados, esperamos 10% do casos. Coletando 20% dos dados esperamos 20% dos casos e assim por diante.

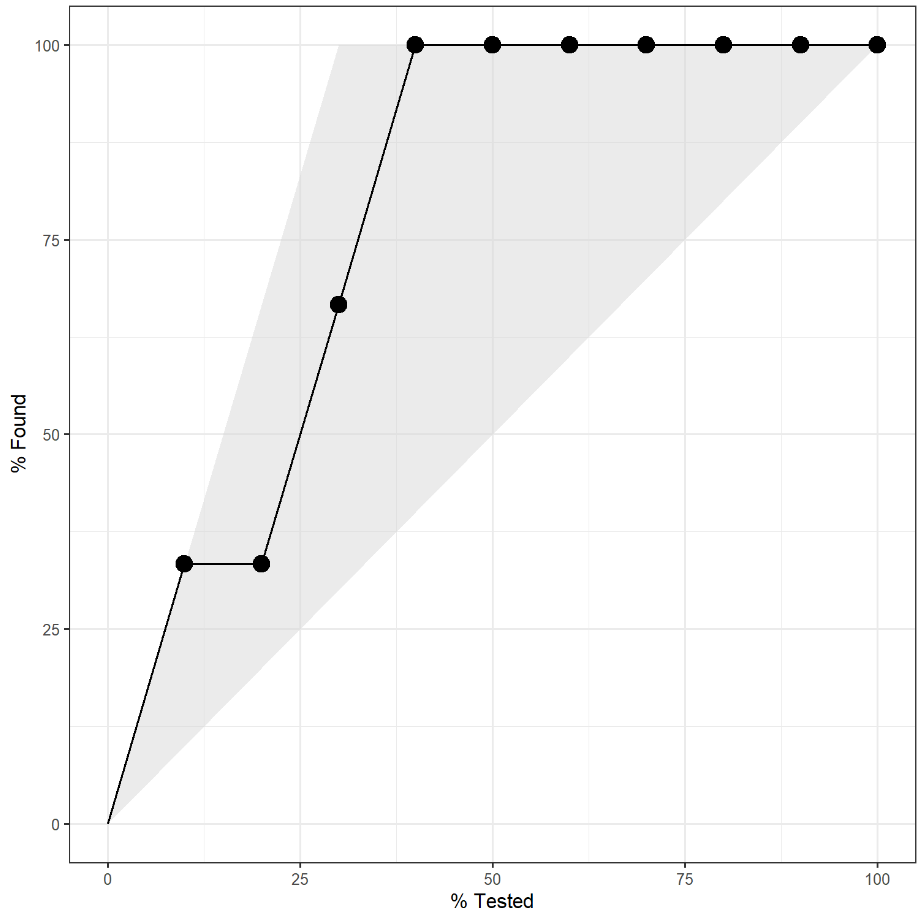
Esse seria o desempenho do modelo nulo/aleatório.



Gain curve

Mostra o quão eficaz o modelo é em identificar casos positivos em relação a uma escolha aleatória (bom para dados desbalanceados).

probabilidade	churn	found	tested
0.95	1	33.33333	10
0.90	0	33.33333	20
0.84	1	66.66667	30
0.82	1	100.00000	40
0.80	0	100.00000	50
0.70	0	100.00000	60
0.60	0	100.00000	70
0.50	0	100.00000	80
0.40	0	100.00000	90
0.30	0	100.00000	100

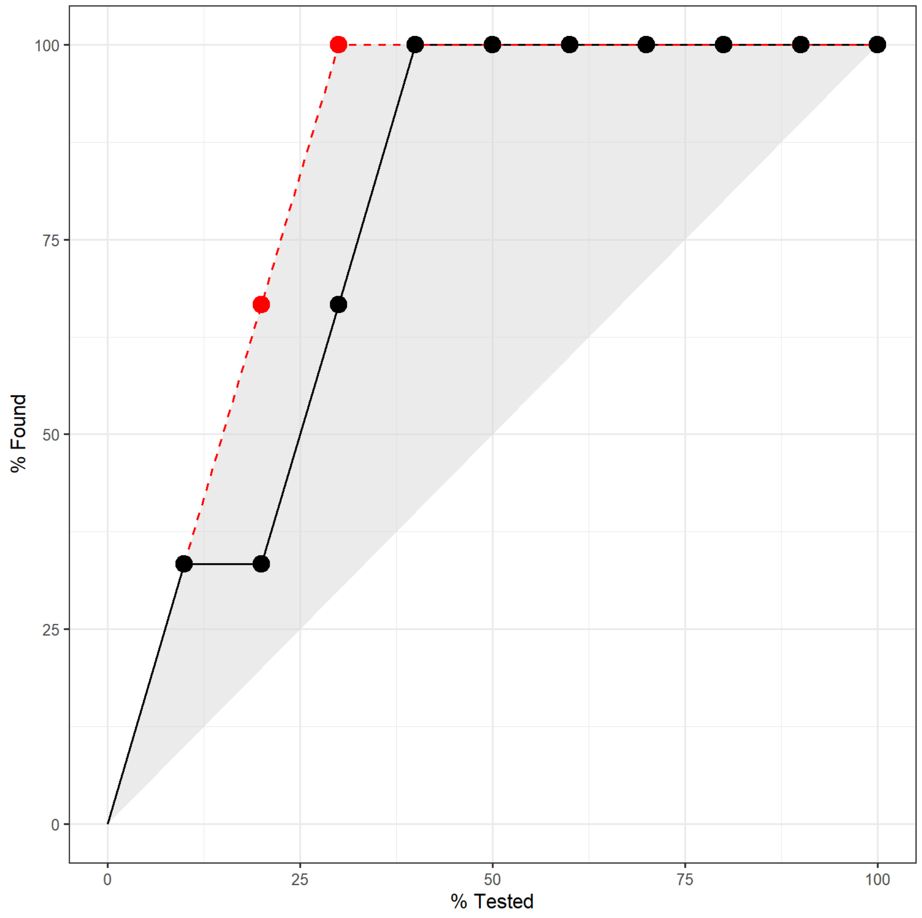


Gain curve

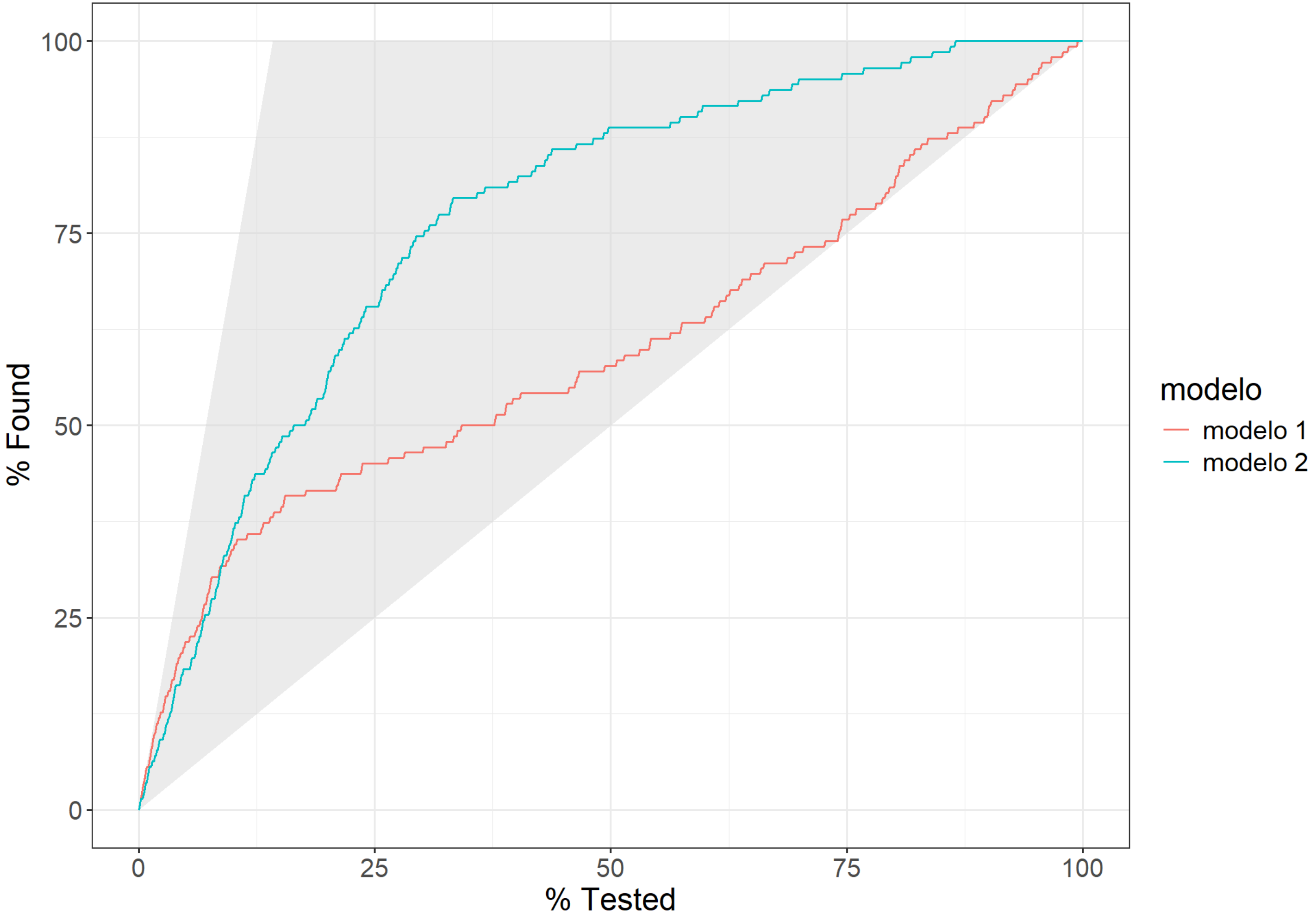
Mostra o quão eficaz o modelo é em identificar casos positivos em relação a uma escolha aleatória (bom para dados desbalanceados).

probabilidade	churn	found	tested
0.95	1	33.33333	10
0.90	0	33.33333	20
0.84	1	66.66667	30
0.82	1	100.00000	40
0.80	0	100.00000	50
0.70	0	100.00000	60
0.60	0	100.00000	70
0.50	0	100.00000	80
0.40	0	100.00000	90
0.30	0	100.00000	100

Os 30% com maiores probabilidades correspondem a 67% do casos. Caso estivéssemos selecionando ao acaso, precisaríamos coletar 67% dos casos para isso. Pense em uma distribuição uniforme.



Gain curve



100
75
50
25
0
% Found
0
25
50
75
100
% Tested
modelo
modelo 1
modelo 2

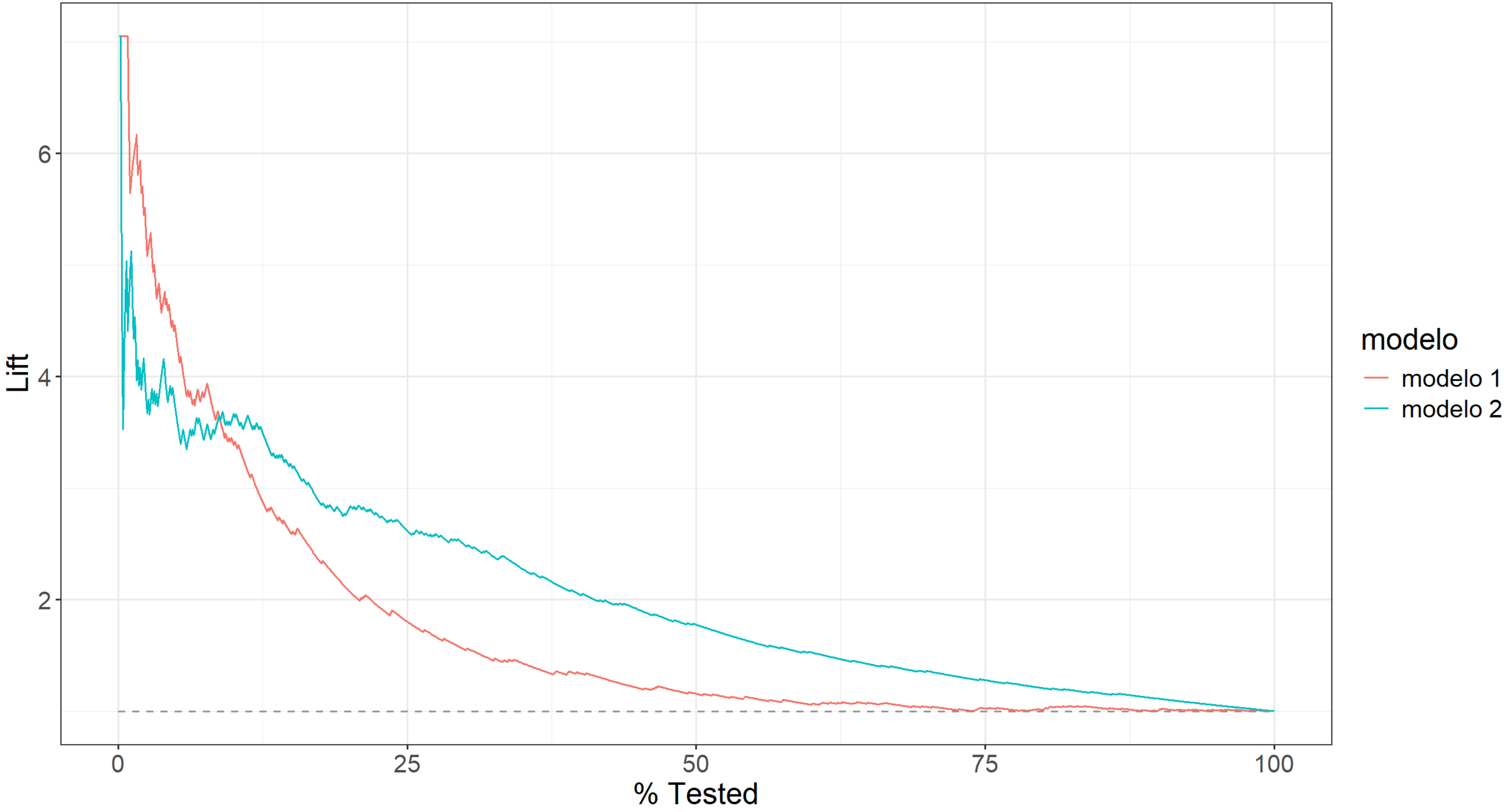
Gain curve

Para gerar a figura do slide anterior, podemos utilizar o seguinte código:

```r
df_avaliacao %>%
  group_by(modelo) %>%
  gain_curve(churn, probabilidade, event_level = "second") %>%
  autoplot()
```

Lift curve

Comparar a **taxa de verdadeiros positivos do modelo** com a taxa de verdadeiros positivos esperada de um **modelo aleatório**.



Lift curve

Para gerar a figura do slide anterior, podemos utilizar o seguinte código:

```
df_avaliacao %>%
  group_by(modelo) %>%
  lift_curve(churn, probabilidade, event_level = "second") %>%
  autoplot()
```

Classificação - Lucro Esperado

Considere os seguintes benefícios:

	Observado	
Classificação	**No**	**Yes**
No	b(No, No) = 10	b(Yes, No) = -20
Yes	b(No, Yes) = -5	b(Yes, Yes) = 100

Entre 100 observações, foi observado o seguinte cenário:

	Observado	
Classificação	**No**	**Yes**
No	30	10
Yes	20	40

$$\text{Lucro Esperado} = 10 \times \frac{30}{100} + (-5) \times \frac{20}{100} + (-20) \times \frac{10}{100} + 100 \times \frac{40}{100} = 40$$

Desafio

A certain disease affects about 1 out of 10,000 people. There is a test to check whether the person has the disease. The test is quite accurate. In particular, we know that

- the probability that the test result is positive (suggesting the person has the disease), given that the person does not have the disease, is only 2 percent;
- the probability that the test result is negative (suggesting the person does not have the disease), given that the person has the disease, is only 1 percent.

A random person gets tested for the disease and the result comes back positive. What is the probability that the person has the disease?

Retirado de https://www.probabilitycourse.com/chapter1/1_4_3_bayes_rule.php.

Dados desbalanceados[1]

Algumas alternativas são consideradas para trabalhar com dados desbalanceados:

- Ajustar o modelo para maximizar a acurácia da classe minoritária
- Escolher o corte para classificação com base na curva ROC
- Ponderar os dados com pesos maiores para as classes minoritárias
- *Down-sampling*: amostra dados da classe majoritária para que tenha a mesma proporção da classe minoritária
- *Up-sampling*: é feito um processo de reamostragem com reposição do grupo minoritário até tenha aproximadamente o mesmo número de observações que o grupo majoritário

[1] *Kuhn, Max and Johnson, Kjell. "Applied predictive modeling." 2013.*

Obrigado!

tiagoms.com

tiagomendonca

tiagoms1@insper.edu.br