Modelos_Estatisticos-2024-08-31

Helena R. S. D'Espindula

2024-08-31

Contents

Modelos lineares generalizados

Prof Cesar Augusto Taconeli

Objetivo

Os modelos lineares generalizados configuram extensões do modelo de regressão linear, permitindo modelar, num contexto de regressão, variáveis respostas com distribuição pertencente à família exponencial de distribuições

Sumário

1 Introdução 2 Família exponencial de distribuições 3 Modelo linear generalizado 4 Estimação 5 Inferência 6 Diagnóstico do ajuste 7 Regressão para dados binários 8 Modelos preditivos 9 Regressão para dados de contagens

Introdução - Modelos Lineares Generalizados (GLM)

- Origem: Nelder e Wedderburn (1972): "Generalized Linear Models", publicado no Journal of the Royal Statistical Society
- Extensão dos modelos lineares, incorporando, sob uma teoria unificada, diversos outros modelos propostos até então.
- Como casos particulares dos modelos lineares generalizados, temos os modelos de regressão linear, a regressão logística para resposta binária e o modelo log-linear para resposta de contagem.

Família exponencial de distribuições

- Os modelos lineares generalizados permitem analisar, num contexto de regressão, variáveis respostas pertencentes à família exponencial de distribuições.
- Mais especificamente, assumimos que a função (densidade) de probabilidades de y possa ser expressa na seguinte forma:

$$f(y;\theta;\phi) = exp\{\frac{y\theta - b\theta}{a(\phi)} + c(y;\phi)\}$$

Revisando

Regressão Linear

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon$$

$$\epsilon \sim N(0, \sigma^2)$$

$$y|x \sim N(\mu_x, \sigma^2)$$

$$E(y|x) = \mu_x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k$$

Modelo Linear Generalizado (GLM)

$$y|x \sim fe(\mu_x, \phi)$$

Sendo fe familia? exponencial de probabilidades

$$g(\mu_x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Sendo $g(u_x)$ a função de ligação

Família exponencial de distribuições

Dentre as principais distribuições pertencentes à família exponencial, temos:

- Binomial (Bernoulli);
- Poisson;
- Normal;

- Gamma (exponencial);
- Normal inversa;
- Binomial negativa*.

$$x \sim N(\mu, \sigma^2)\{E(x) = \mu; Var(x) = \sigma^2$$
$$y \sim Poisson(\mu)\{E(x) = \mu; Var(x) = \mu$$

$$z \sim Bernoilli(\pi)\{E(x) = \pi; Var(x) = \pi(1 - \pi)$$

Distribuição binomial

A distribuição binomial é uma alternativa na modelagem de dados binários (dicotômicos). Como exemplos:

- E-mails classificados por um algoritmo como spam ou não spam;
- Clientes de um banco classificados como pagadores ou não pagadores;
- Pacientes submetidos a certo tipo de cirurgia que apresentam ou não determinada sequela;
- Resultados dos jogos da NBA (liga norte-americana de basquete) quanto à vitória ou derrota do time mandante.

[. . .]

Definição de um modelo linear generalizado

- Um modelo linear generalizado é definido pela especificação de três componentes: o componente aleatório, o componente sistemático e uma função de ligação.
- Componente aleatório: Uma variável aleatória (resposta) com distribuição pertencente à família exponencial.
- Como vimos anteriormente, são membros dessa família as distribuições binomial, Poisson, normal, gama, normal inversa. . .
- Um modelo linear generalizado é definido da seguinte forma:

$$y|x \sim f(\mu_x, \phi),$$

em que $f(\cdot)$ representa alguma particular distribuição pertencente à família exponencial, e

$$g(\mu_x) = \eta x = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

em que $g(\cdot)$ é uma função real que desempenha a ligação dos componentes aleatório e sistemático do modelo.

Componentes de um modelo linear generalizado

- Componente sistemático: preditor linear do modelo, em que são inseridas as covariáveis $(x_1, x_2, ..., x_p)$ e um conjunto de parâmetros associados em uma função linear.

$$\eta x = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

- Função de ligação: Função real, monótona e diferenciável, denotada por $g(\cdot)$, que conecta os componentes aleatório e sistemático do modelo.

Seja μ = E (Y |x1, x2, . . . , xp). Então:

$$g(\mu) = \eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p.$$

O modelo pode ser escrito de maneira equivalente por:

$$\mu = g - 1(\eta) = g - 1(\beta_0 + \beta_1 x_i + \beta_2 x_i + ... + \beta_p x_i$$

Especificação do componente aleatório

- Requer a definição de uma distribuição de probabilidades para a variável resposta.
- A variável resposta é discreta ou contínua? Sua distribuição é simétrica? Qual o conjunto de valores com probabilidade não nula?
- Deve-se propor uma distribuição que tenha propriedades compatíveis aos dados.
- Não se tendo convicção sobre uma particular escolha, pode-se testar diferentes alternativas ou usar alguma abordagem que não exija essa especificação.
- Quais variáveis explicativas devem ser consideradas?
- Como essas variáveis serão incorporadas ao modelo? Avaliar a necessidade (conveniência) de escalonar, transformar, categorizar ou incluir potências de variáveis numéricas. . .
- Avaliar a necessidade de incluir efeitos de interação entre variáveis.

Especificação da função de ligação

- A função de ligação tem o papel de linearizar a relação entre os componentes aleatório e sistemático do modelo
- Deve produzir valores válidos para μ (ou π, no caso da distribuição binomial) para qualquer valor de

$$\eta x = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

  Proporcionar interpretações práticas para os parâmetros de regressão β's.

Exemplo Auditoria

```
#############################################################################
### Dados sobre auditoria na prestação de contas de 3000 indivíduos.

### As variáveis são as segintes:
### valor: é  o valor da nota (em milhares de dólares)

### resultado: e a resposta é o resultado da auditoria (1, se a nota tem indício de fraude
### e 0, não tinha indícios). O objetivo é modelar o resultado da auditoria em função
### do valor da nota.

auditoria <- read.csv2('auditoria.csv')
head(auditoria, 25)
```

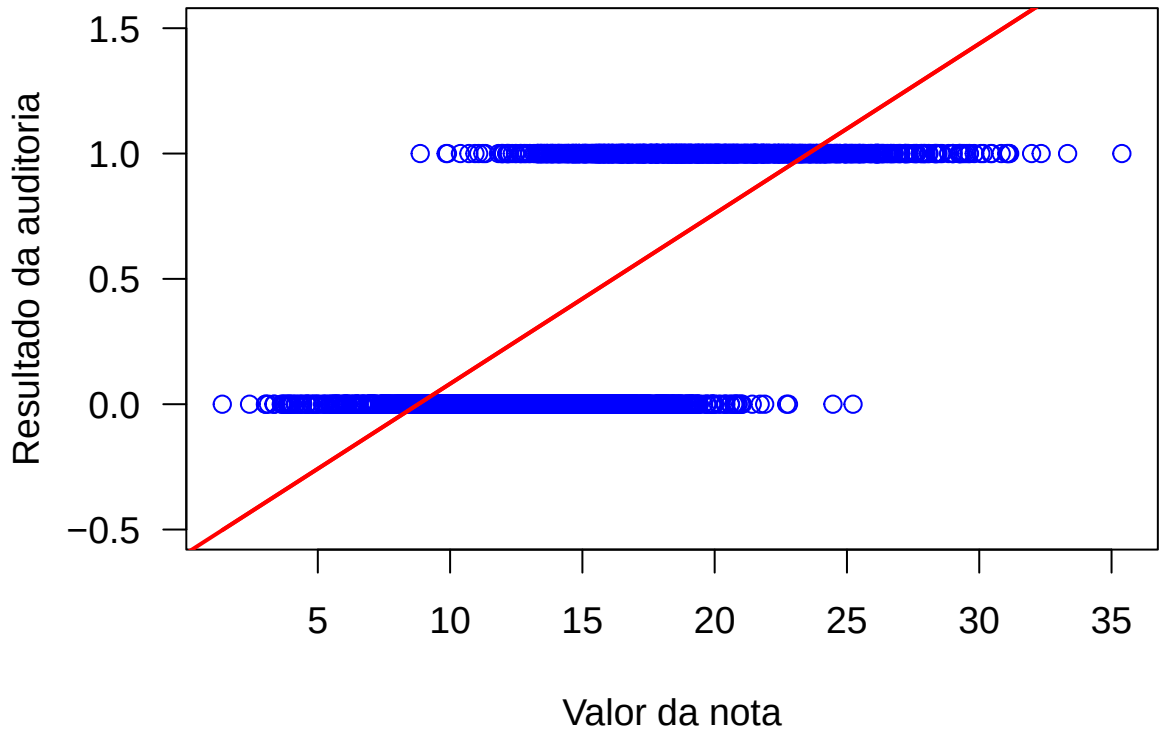
```
##    resultado valor
## 1          0  8.52
## 2          1 18.01
## 3          1 17.58
## 4          0 13.94
## 5          0 17.00
## 6          0 13.00
## 7          0 12.81
## 8          0 10.45
## 9          0  7.49
## 10         0 12.22
## 11         0 18.16
## 12         1 19.75
## 13         0 10.69
## 14         1 26.04
## 15         0  8.49
## 16         1 17.98
## 17         1 12.66
## 18         1 18.41
## 19         1 16.74
## 20         1 19.83
## 21         0 15.34
## 22         1 20.24
## 23         1 16.16
## 24         0 13.85
## 25         0  9.63
```

```r
options(device = 'x11')
```

```r
par(las = 1, mar = c(5,4,2,2), cex = 1.2)
plot(resultado ~ valor, data = auditoria, ylim = c(-0.5, 1.5), col = 'blue',
     xlab = 'Valor da nota', ylab = 'Resultado da auditoria')
### Aparentemente, notas de maior valor estão mais propensas a indícios de fraude.

### Vamos ajustar um modelo de regressão linear.
ajuste <- lm(resultado ~ valor, data = auditoria)

abline(coef(ajuste), col = 'red', lwd = 2)
lines(sort(auditoria$valor), fitted(ajuste)[order(auditoria$valor)], col = 'red', lwd = 2)
```




```
### O modelo ajustado claramente não é apropriado. Observe que para determinados
### valores da variável explicativa, temos valor ajustado inferior a zero
### ou superior a 1.
```

$$\pi = P(y = 1) = P(fraude)$$

Regressão Linear:

$$\pi = \beta_0 + \beta_1 \times valor$$

Função de Ligação: Logito

$$ln(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 \times valor$$

$$\frac{\pi}{1-\pi} = e^{\beta_0 + \beta_1 \times valor}$$

$$\pi = \frac{e^{\beta_0 + \beta_1 \times valor}}{e^{\beta_0 + \beta_1 \times valor} + 1}$$

Tudo isso para ficar entre 0 e 1....

$$log(\frac{\hat{\pi}}{1-\hat{\pi}}) = -9,42 + 0,58 \times valor$$

$$\frac{\hat{\pi}}{1-\hat{\pi}} = e^{-9{,}42+0{,}58\times valor}$$

$$\hat{\pi} = \frac{e^{-9{,}42+0{,}58\times valor}}{e^{-9{,}42+0{,}58\times valor}+1}$$

```
### Vamos contornar isso ajustando um modelo com resposta binomial, o que
### permitirá modelar a probabilidade de uma conta apresentar erros condicional
### ao valor da nota. Vamos avaliar diferentes funções de
### ligação que podem ser usadas na modelagem de dados binários.

par(las = 1, mar = c(5,4,2,2), cex = 1.2)
plot(resultado ~ valor, data = auditoria, pch = "|", ylim = c(0,1), col = 'lightblue',
     xlab = 'Valor da nota', ylab = 'Resultado da auditoria')


ajuste2 <- glm(resultado ~ valor, family = binomial(link = logit), data = auditoria)
### Ligação logito.
summary(ajuste2)
```

```
##
## Call:
## glm(formula = resultado ~ valor, family = binomial(link = logit),
##     data = auditoria)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.42164    0.34874  -27.02   <2e-16 ***
## valor        0.58420    0.02131   27.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4156.2  on 2999  degrees of freedom
## Residual deviance: 2163.5  on 2998  degrees of freedom
## AIC: 2167.5
##
## Number of Fisher Scoring iterations: 6
```

```
lines(sort(auditoria$valor), predict(ajuste2, type = 'response')[order(auditoria$valor)],
      col = 'black', lwd = 2)
### Adicionando a regressão ajustada ao gráfico.


ajuste3 <- glm(resultado ~ valor, family = binomial(link = probit), data = auditoria)
### Ligação probito.
summary(ajuste3)
```

```
##
## Call:
## glm(formula = resultado ~ valor, family = binomial(link = probit),
```

```
##     data = auditoria)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.34183    0.17835  -29.95   <2e-16 ***
## valor        0.33095    0.01085   30.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4156.2  on 2999  degrees of freedom
## Residual deviance: 2164.7  on 2998  degrees of freedom
## AIC: 2168.7
##
## Number of Fisher Scoring iterations: 6
```

```
lines(sort(auditoria$valor), predict(ajuste3, type = 'response')[order(auditoria$valor)],
      col = 'red', lwd = 2)
```

```
ajuste4 <- glm(resultado ~ valor, family = binomial(link = cloglog), data = auditoria)
```

```
## Warning: glm.fit: probabilidades ajustadas numericamente 0 ou 1 ocorreu
```

```
### Ligação complemento log-log.
summary(ajuste4)
```

```
##
## Call:
## glm(formula = resultado ~ valor, family = binomial(link = cloglog),
##     data = auditoria)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.02785    0.20621  -29.23   <2e-16 ***
## valor        0.34154    0.01172   29.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4156.2  on 2999  degrees of freedom
## Residual deviance: 2239.1  on 2998  degrees of freedom
## AIC: 2243.1
##
## Number of Fisher Scoring iterations: 8
```

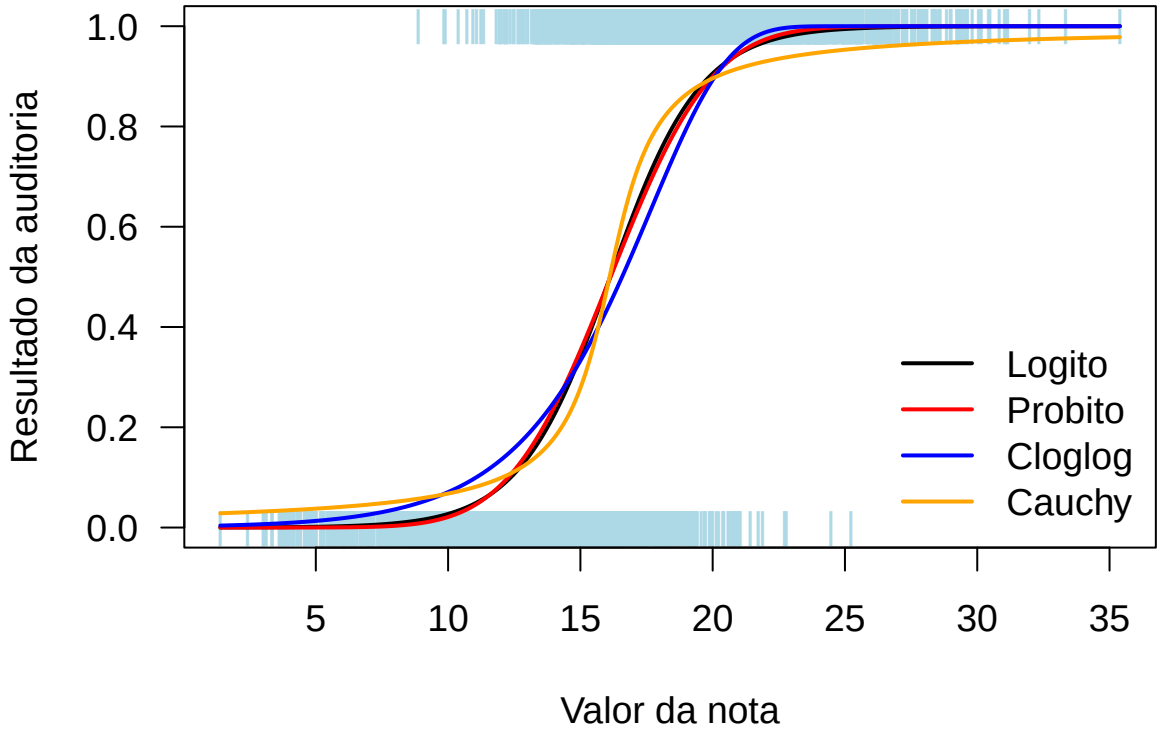
```
lines(sort(auditoria$valor), predict(ajuste4, type = 'response')[order(auditoria$valor)],
      col = 'blue', lwd = 2)
```

```r
ajuste5 <- glm(resultado ~ valor, family = binomial(link = cauchit), data = auditoria)
### Ligação Cauchy.
summary(ajuste5)
```

```
##
## Call:
## glm(formula = resultado ~ valor, family = binomial(link = cauchit),
##     data = auditoria)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.2010     0.7563  -16.13   <2e-16 ***
## valor         0.7576     0.0468   16.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4156.2  on 2999  degrees of freedom
## Residual deviance: 2233.1  on 2998  degrees of freedom
## AIC: 2237.1
##
## Number of Fisher Scoring iterations: 6
```

```r
lines(sort(auditoria$valor), predict(ajuste5, type = 'response')[order(auditoria$valor)],
      col = 'orange', lwd = 2)

legend(x = 'bottomright', lwd = 2, col = c('black', 'red', 'blue', 'orange'),
       legend = c('Logito', 'Probito', 'Cloglog', 'Cauchy'), bty = 'n')
```




```
### Aparentemente, os modelos com ligação logito e probito proporcionam melhor
### ajuste que os demais. Além disso, os ajustes desses dois modelos são
### bastante semelhantes. Vamos comparar os modelos com base nos respectivos
### AICs.
```

```
AIC(ajuste2, ajuste3, ajuste4, ajuste5)
```

```
##         df      AIC
## ajuste2  2 2167.462
## ajuste3  2 2168.702
## ajuste4  2 2243.110
## ajuste5  2 2237.070
```

```
### O ajuste 2 (modelo com ligação logito) produziu menor AIC, sendo preferível.
```

Exemplo Credito

```
################################################################################
################################################################################
################################################################################
### Regressão para dados binários - modelo preditivo.
### Dados sobre concessão de crédito. O objetivo é modelar a variável
### resposta (default), que indica se o indíviduo atrasou (Yes) ou não (No)
### o pagamento da fatura do cartão de crédito. As variáveis explicativas
```

```
### são as seguintes:

### income: renda anual;
### balance: saldo devedor no cartão de crédito no último período;
### student: Yes, para estudante; No, caso contrário.

### Carregando os pacotes necessários para a análise.
require(ISLR)
require(statmod)
require(pROC)
require(car)
require(hnp)

### Carregamento e preparação dos dados.
data("Default")
help("Default")
summary(Default)
Default$income <- Default$income/1000 ### Renda em x$1.000.

### Podemos observar que a frequência de devedores é bastante inferior à de
### não devedores.


################################################################################
################################################################################
################################################################################
### Análise exploratória.

ggplot(data = Default, aes(x = student, group = default, fill = default)) +
    geom_bar(stat = 'count', position = position_dodge())+
    theme_bw(base_size = 14)+
    theme(legend.position = 'bottom')+
    geom_text(aes( y=..count.., label=scales::percent(..count../tapply(..count.., ..x.. ,sum)[..x..], a
              stat="count", position=position_dodge(0.9), vjust=-0.5)

ggplot(data = Default, aes(x = default, y = balance, fill = default)) +
    geom_boxplot()+
    theme_bw(base_size = 14)+
    theme(legend.position = 'none')

ggplot(data = Default, aes(x = default, y = income, fill = default)) +
    geom_boxplot()+
    theme_bw(base_size = 14)+
    theme(legend.position = 'none')

### Separação da base de dados em dados de treino e dados de teste
set.seed(2024)
w <- sample(1:nrow(Default), size = 7000)

dados_treino <- Default[w,]
dados_teste <- Default[-w,]

################################################################################
################################################################################
```

```
##############################################################################
### Ajuste do modelo de regressão logística

ajuste <- glm(default ~ ., data = dados_treino, family = binomial(link = 'logit'))
options(scipen = 5)
summary(ajuste)

### Algumas interpretações:
### A log-chance de default aumenta em 0.0057 para $1 a mais no
### balanço. De maneira equivalente, a chance de não pagamento fica
### multiplicada por exp(0.0057) = 1.006 para $1 a mais no balanço,
### ou ainda, por exp(100*0.0057) = 1.768 para $100 a mais no balanço
### (fixados os valores das demais covariáveis).

### Alog-chance de não pagamento para estudantes é 0.725 menor do que para não
### estudantes, ou ainda, a chance de não pagamento fica multiplicada
### por exp(-0.631) = 0.53 para estudantes em relação a não estudantes,
### fixados os valores das demais covariáveis.

### O efeito de renda, ajustado pelas outras duas variáveis, não é significativo.

##############################################################################
### Análise de resíduos

### Vamos dar sequência à análise com o diagnóstico do ajuste.
par(mfrow = c(2,2))
plot(ajuste)
### Os gráficos de resíduos têm comportamento bastante atípico, mas característico
### da análise de dados binários, devido aos empates. Para um diagnóstico mais
### adequado, vamos usar os resíduos quantílicos aleatorizados, disponíveis
### no pacote statmod, e os gráficos meio-normais com envelope simulado,
### disponíveis no pacote hnp.

residuos <- qres.binom(ajuste)
ajustados <- predict(ajuste)
# x11(width = 15, height = 10)
par(las = 1, mar = c(5,4.5,2,2), mfrow = c(1,2), cex = 1.2)
plot(residuos ~ ajustados, col = 'blue', xlab = 'Valores ajustados', ylab = 'Resíduos')
lines(lowess(residuos ~ ajustados), col = 'red', lwd = 2)
qqnorm(residuos, col = 'blue', main = '')
qqline(residuos, lty = 2)

### Os resíduos apresentam dispersão aleatória, variância aprox. constante e
### distribuição normal. O modelo parece estar bem ajustado.

par(las = 1, mar = c(5,4.5,2,2), cex = 1.4)
hnp(ajuste)
### O padrão para um ajuste adequado é os resíduos (pontos) dispostos no
### interior do envelope (linhas) simulado. Novamente temos um indicativo
### de que o modelo está bem ajustado.

##############################################################################
##############################################################################
##############################################################################
```

```
### Inferência estatística e redefinição do modelo

### Como o efeito de renda não se mostrou significativo, vamos removê-lo
### do modelo.
ajuste2 <- update(ajuste, ~.-income)
summary(ajuste2)

### Agora, vamos usar o modelo ajustado para fins de predição. Antes de
### utilizar a base de validação, vamos fazer predição para algus dados
### adicionais.
novos_dados <- data.frame(student = rep(c('Yes', 'No'), times = 3),
                          balance = c(500, 500, 1000, 1000, 1750, 1750))
### Base para predição.

exp(predict(ajuste2, newdata = novos_dados))


exp(predict(ajuste2, newdata = novos_dados))
### Predição na escala do preditor (log-chance de default)

predict(ajuste2, newdata = novos_dados, type = 'response')
### Predição na escala da resposta (probabilidade de default, inversa do link)

### Agora, vamos fazer intervalos de confiança (95%) para a probabilidade
p_link <- predict(ajuste2, newdata = novos_dados, se.fit = TRUE)
### Predições na escala do link com os erros padrões associados.

ic_link <- cbind(p_link$fit - 1.96 * p_link$se.fit, p_link$fit + 1.96 * p_link$se.fit)
ic_link
### Intervalos de confiança (95%) para a log-chance de default.

exp(ic_link)/(exp(ic_link) + 1)
### Intervalos de confiança (95%) para a probabilidade de default.

##############################################################################
##############################################################################
##############################################################################
### Validação do modelo usando a base de teste.

### na sequência, vamos retomar a amostra de validação para avaliar a
### capacidade preditiva do modelo.

predicoes <- predict(ajuste2, newdata = dados_teste, type = 'response')
### Probabilidades estimadas de default para os indivíduos da base de validação.

hist(predicoes, breaks = 20, main = '')

### Vamos ver como ficaria o resultado da predição se adotássemos o ponto
### de corte p=0.5 para predição, isto é, classificando como não pagadores
### os indivíduos com probabilidade estimada superior a 0.5 e como pagadores
### aqueles com probabilidade inferior a 0.5.

tab_pred <- table(ifelse(predicoes < 0.5, 'Pred_No', 'Pred_Yes'), dados_teste$default)
tab_pred
```

```
prop.table(tab_pred, 2)

### A regra de classificação baseada no ponto de corte p = 0.5 tem elevada
### especificidade (0.995), mas baixa sensibilidade (0.361). Neste
### problema, em particular, sensibilidade (identificar não pagadores)
### deve ser mais importante que especificidade (identificar pagadores).
### Desta forma, poderíamos considerar um valor menor para o ponto de corte,
### visando aumentar a sensibilidade do modelo. Vejamos como ficariam os
### resultados para p = 0.1.

tab_pred <- table(ifelse(predicoes < 0.1, 'Pred_No', 'Pred_Yes'), dados_teste$default)
tab_pred
prop.table(tab_pred, 2)

### Neste cenário, a especificidade é ligriramente reduzida para 0.946.
### Em contrapartida, a sensibilidade é aumentada para 0.771.
### Nesse sentido, precisamos explorar adequadamente a capacidade preditiva
### do modelo e buscar regras de classificação alternativas. Vamos usar
### os recurdos do pacote pRoc.

r1 <- roc(dados_teste$default, predicoes, plot=TRUE, ci=TRUE, ci.sp = TRUE)
r1
### A área sob a curva ROC é uma medida de qualidade preditiva do modelo.
### Valores próximos de 1 indicam modelos com elevada capacidade preditiva,
### enquanto valores próximos de 0.5 indicam modelos cujas predições são
### realizadas ao acaso. Mais do que interpretá-lo, é um indicador importante
### para comparação da performance de diferentes modelos preditivos aplicados
### a uma base de dados.

plot(r1, print.thres = c(0.001, 0.005, 0.01, 0.02, 0.03, 0.04, seq(0.05,0.95,0.05)),
     print.thres.pattern.cex = 0.8)
### Curva ROC. O valor que aparece fora dos parênteses é o ponto de corte.
### No interior temos a sensibilidade e a especificidade correspondentes,
### respectivamente. Pontos de corte posicionados no canto superior esquerdo
### são aqueles que combinam maior sensibilidade e especificidade. Pontos
### de corte em torno de 0.05 produzem regras de classificação que conjugam
### elevadas sensibilidade e especificidade. No entanto, aspectos operacionais
### e do relacionamento com os clientes também devem ser levados em consideração
### ao se estabelecer a regra de classificação.

coords(r1, x = 0.01, ret = c("sensitivity", "specificity", "accuracy"))
### Sensibilidade, especificidade e acurácia para o ponto de corte p = 0.01.

coords(r1, x = 0.05, ret = c("sensitivity", "specificity", "accuracy"))
### Sensibilidade, especificidade e acurácia para o ponto de corte p = 0.05.

### Agora, vamos identificar a melhor regra de decisão (ponto de corte)
### associada a diferentes custos de máclassificação. No argumento
### "best.weights=c(a, b)", em que "a" representa o custo de um falso negativo
### relativo a um falso positivo e "b" a prevalência (proporção de sucessos)
### na população.

### Vamos lembrar que, neste exemplo, falso negativo corresponde a classificar
### como pagador um não pagador. A prevalência de maus pagadores nós vamos
```

```
### fixar em 0.033, que é a prevalência verificada na base.

coords(r1, x = "best", best.method = "youden", best.weights=c(1, 0.033))
### Custos iguais.

coords(r1, x = "best", best.method = "youden", best.weights=c(2, 0.033))
### O custo do falso negativo é duas vezes o do falso positivo.

coords(r1, x = "best", best.method = "youden", best.weights=c(5, 0.033))
### O custo do falso negativo é cinco vezes o do falso positivo.

coords(r1, x = "best", best.method = "youden", best.weights=c(20, 0.033))
### O custo do falso negativo é vinte vezes o do falso positivo.

### Na área de concessão de crédito, é usual se trabalhar com o chamado
### "escore de crédito", que é 100*P(pagador|x). Ou seja:

cred_escores <- 100*(1-predicoes)
head(cred_escores, 20)
hist(cred_escores, main = '')
```