Modelos Lineares Generalizados

Guilherme Rodrigues

2/2019

Exemplo: notas em PE

A disciplina de Probabilidade e Estatistica (PE) da UnB adota um sistema de avaliação automatizado usando o pacote `R exams`. Neste exemplo, usaremos os dados do primeiro semestre de 2019 para modelar o efeito do resultado da primeira prova na nota final do aluno no curso (desconsiderando-se a prova substitutiva).

Análise exploratória dos dados

Carregando o banco de dados e os pacotes

```r
library(tidyverse)
library(reshape2)
library(broom)
dados.originais <- read.csv2("Banco_respostas.csv")
str(dados.originais)
```

```
## 'data.frame':    13905 obs. of  14 variables:
##  $ Matricula   : int  120007789 140023992 140137734 150048530 1500
##  $ Ano         : int  2012 2014 2014 2015 2015 2016 2016 2016 2016
##  $ Turma       : Factor w/ 10 levels "AA","AB","BA",..: 7 7 7 7 7
##  $ Horario     : Factor w/ 5 levels "08/out","10/dez",..: 3 3 3 3
##  $ Professor   : Factor w/ 7 levels "ANDRE LUIZ","EDUARDO YOSHIO",
##  $ Codigo_curso: int  60844 6254 6254 3221 8150 6335 3221 1589 635
```

Carregando o banco de dados e os pacotes

```r
library(tidyverse)
library(reshape2)
library(broom)
dados.originais <- read.csv2("Banco_respostas.csv")
str(dados.originais)
```

```
## 'data.frame':    13905 obs. of  14 variables:
##  $ Matricula   : int  120007789 140023992 140137734 150048530 1500
##  $ Ano         : int  2012 2014 2014 2015 2015 2016 2016 2016 2016
##  $ Turma       : Factor w/ 10 levels "AA","AB","BA",..: 7 7 7 7 7
##  $ Horario     : Factor w/ 5 levels "08/out","10/dez",..: 3 3 3 3
##  $ Professor   : Factor w/ 7 levels "ANDRE LUIZ","EDUARDO YOSHIO",
##  $ Codigo_curso: int  60844 6254 6254 3221 8150 6335 3221 1589 635
```

Tranformando os dados

```
indices <- match(unique(dados.originais$Matricula),
                 dados.originais$Matricula)
dados <- dados.originais %>%
  filter(Numero.prova <= 3) %>%
  group_by(Matricula, Numero.prova) %>%
  summarize(Nota_prova = sum(Acertou)) %>%
  dcast(Matricula ~ Numero.prova, value.var="Nota_prova") %>%
  mutate_all(list(~replace_na(., 0))) %>%
  rename(P1=2, P2=3, P3=4) %>%
  mutate(Nota_final=(.3*P1 + .3*P2 + .4*P3),
         Aprovado=Nota_final>=5) %>%
  left_join(dados.originais[indices, ], by="Matricula")
```

Dados Transformados

```
str(dados)
```

```
## 'data.frame':    464 obs. of  19 variables:
##  $ Matricula   : num  1.0e+08 1.2e+08 1.2e+08 1.2e+08 1.2e+08 ...
##  $ P1          : num  0 7 0 7 4 2 4 5 3 0 ...
##  $ P2          : num  7 3 5 7 0 6 0 4 5 0 ...
##  $ P3          : num  6 5 8 6 2 3 0 3 2 0 ...
##  $ Nota_final  : num  4.5 5 4.7 6.6 2 3.6 1.2 3.9 3.2 0 ...
##  $ Aprovado    : logi  FALSE TRUE FALSE TRUE FALSE FALSE ...
##  $ Ano         : int  2010 2012 2012 2012 2012 2013 2014 2014 2014
##  $ Turma       : Factor w/ 10 levels "AA","AB","BA",..: 4 7 5 1 4
##  $ Horario     : Factor w/ 5 levels "08/out","10/dez",..: 2 3 3 1
##  $ Professor   : Factor w/ 7 levels "ANDRE LUIZ","EDUARDO YOSHIO",
```

Dados Transformados

```
str(dados)
```

```
## 'data.frame':    464 obs. of  19 variables:
##  $ Matricula   : num  1.0e+08 1.2e+08 1.2e+08 1.2e+08 1.2e+08 ...
##  $ P1          : num  0 7 0 7 4 2 4 5 3 0 ...
##  $ P2          : num  7 3 5 7 0 6 0 4 5 0 ...
##  $ P3          : num  6 5 8 6 2 3 0 3 2 0 ...
##  $ Nota_final  : num  4.5 5 4.7 6.6 2 3.6 1.2 3.9 3.2 0 ...
##  $ Aprovado    : logi  FALSE TRUE FALSE TRUE FALSE FALSE ...
##  $ Ano         : int  2010 2012 2012 2012 2012 2013 2014 2014 2014
##  $ Turma       : Factor w/ 10 levels "AA","AB","BA",..: 4 7 5 1 4
##  $ Horario     : Factor w/ 5 levels "08/out","10/dez",..: 2 3 3 1
##  $ Professor   : Factor w/ 7 levels "ANDRE LUIZ","EDUARDO YOSHIO",
```

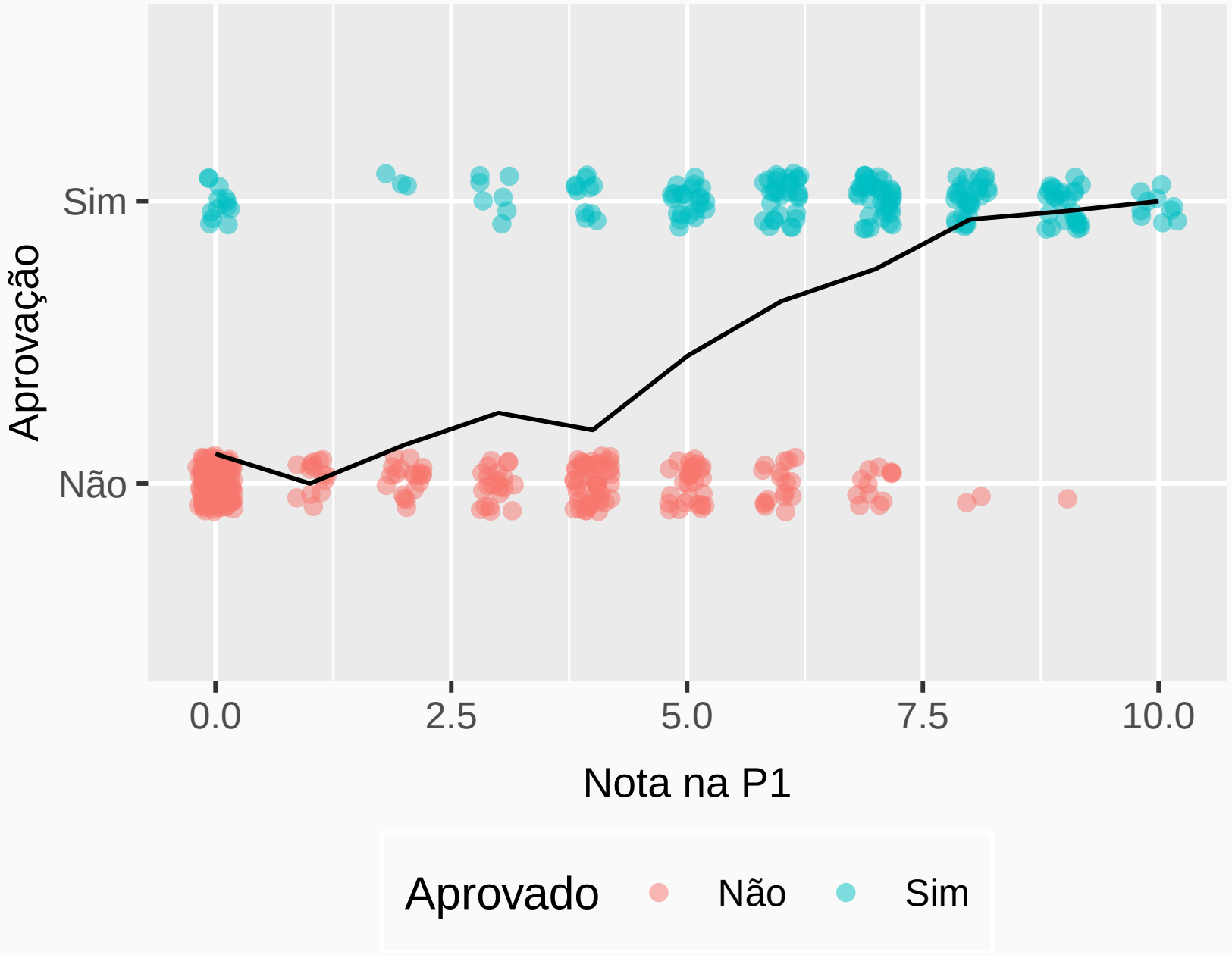
Sumarizando os dados de acordo com a prova 1

```
dados.agrupados <- dados %>%
  group_by(P1) %>%
  summarise(n=n(),
            reprovados=n() - sum(Aprovado),
            aprovados=sum(Aprovado),
            taxa=mean(Aprovado))
dados.agrupados
```

```
## # A tibble: 11 x 5
##       P1     n reprovados aprovados  taxa
##    <dbl> <int>      <int>     <int> <dbl>
##  1     0   124        111        13 0.105
##  2     1    14         14         0 0
```

Plotando o gráfico de Y em X

```
grafico.y <- ggplot(dados) +
  labs(y="Aprovação", x="Nota na P1") +
  geom_jitter(aes(P1, as.numeric(Aprovado), color=Aprovado),
              height=.1, width=.2, alpha=.5) +
  scale_color_discrete(labels=c("Não", "Sim")) +
  geom_line(data=dados.agrupados, aes(P1, taxa), lwd=.5) +
  scale_y_discrete(breaks=c(0, 1),
                   labels=c("Não", "Sim"),
                   limits=c(0, 1))
```


Aprovação
Sim
Não
0.0
2.5
5.0
7.5
10.0
Nota na P1
Aprovado
Não
Sim

Definindo o modelo

Definindo o Modelo Logístico

Distribuição das observações: $Y_i \overset{ind.}{\sim} \text{Bernoulli}(p_i)$

Função de ligação logito:

$$g(p_i) = \log\left(\frac{p_i}{1-p_i}\right)$$

$$= \eta_i = \boldsymbol{x}_i'\boldsymbol{\beta} = \sum_{j=1}^{p} x_{ji}\beta_j,$$

onde η_i é o preditor linear (relacionado ao indivíduo i), x_{ji} é o valor da covariável j associada ao indivíduo i (fixa e conhecida) e $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)'$ é um vetor de parâmetros desconhecidos. Neste exemplo, iremos considerar $\beta = (\beta_0, \beta_1)$, onde β_0 é o intercepto e β_1 o o efeito da Prova 1 nota na nota final do aluno.

Suposições (parte 1)

- O modelo assume que as notas dos alunos são independentes umas das outras. Isso não parece razoável, uma vez que os alunos estão agrupados em turmas. Alternativas: incorporar a informação da turma entre as covariáveis ou adotar modelos hierarquicos (modelos mixtos).
- Como apenas as notas na P1 foram consideradas no modelo, os alunos de um mesmo curso também terão notas correlacionadas. Alternativas: incluir variáveis faltantes.

Suposições (parte 2)

- A probabilidade de aprovação cresce (ou decresce) monotonicamente em função da nota na P1. É possível que isso não seja razoável. Alternativas: incluir outros termos polinomiais ou adotar modelos aditivos generalizados (GAN).
- A função de ligação logito é adequada. É importante avaliar se outras opções (probito, por exemplo) resulta em um modelo de maior qualidade.
- Pode-se modelar a aprovação indiretamente, modelando-se a nota final dos alunos.

Estimador de máxima verossimilhança (EMV)

Função logistica e Log-verossimilhança

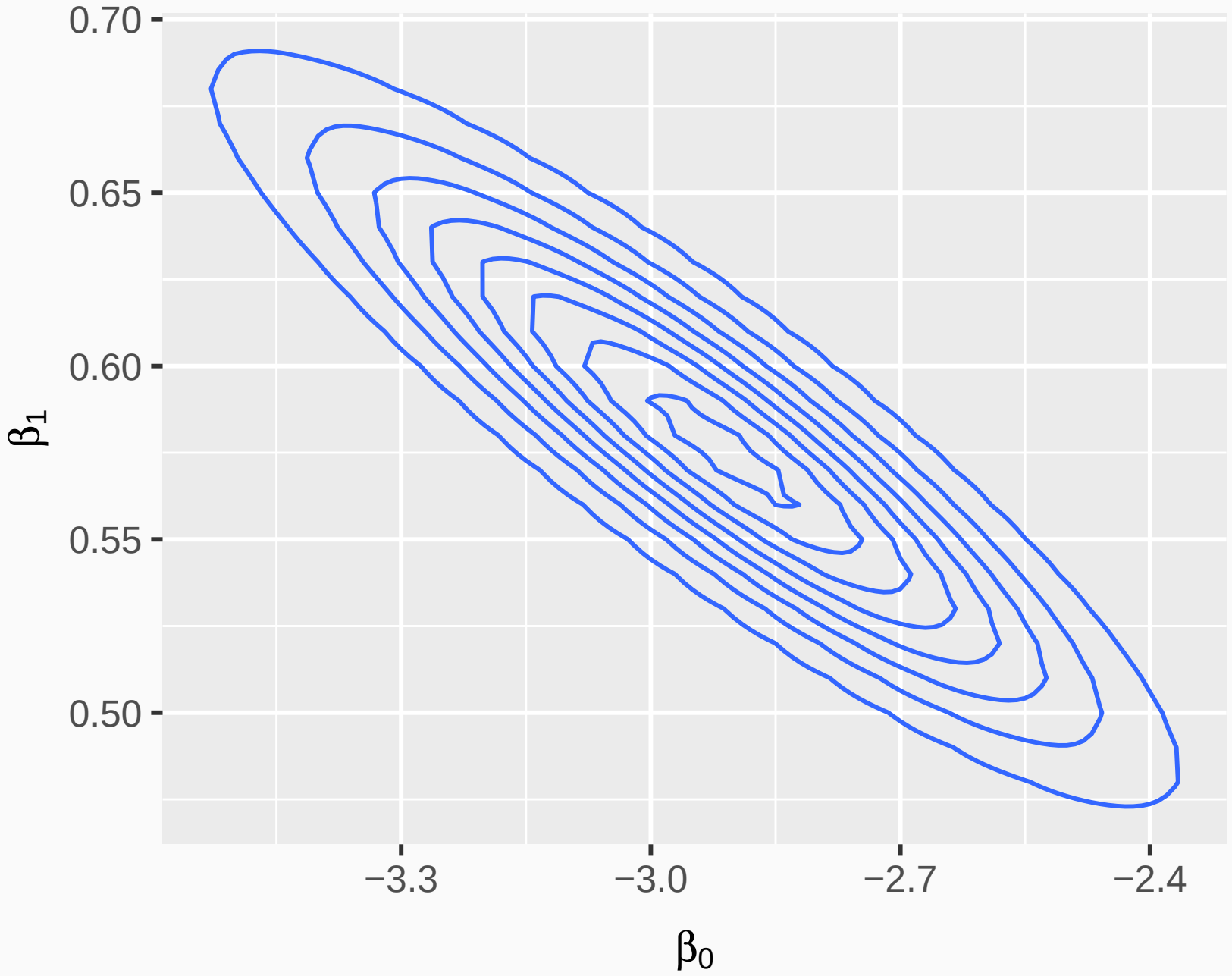
```r
# Função logistica (inverso da logito)

logistic <- function(eta) exp(eta) / (1 + exp(eta))


# Log-verossimilhança

log.L <- function(beta, x, y) {
  eta <- x %*% beta
  p <- logistic(eta)
  sum(dbinom(y, 1, p, log=T))
}
```

Plotando a log-verossimilhança em um grid (parte 1)

```
X <- cbind(1, dados$P1)  # Matriz de delineamento
Y <- dados$Aprovado # Vetor das observações
beta.grid <-  expand.grid(beta0.grid = seq(-5, 0, .01),
  beta1.grid = seq(0, 1, .01))
log.L.grid <- apply(beta.grid, 1,
                    function(.) log.L(., x=X, y=Y))
data.grid <- cbind(beta.grid, log.L.grid)
```

Plotando a log-verossimilhança em um grid (parte 2)

```
contorno <- ggplot(data.grid,
                   aes(beta0.grid,
                       beta1.grid,
                       z=exp(log.L.grid))) +
  geom_contour() +
  labs(y=expression(beta[1]), x=expression(beta[0]))
```


0.70
0.65
0.60
0.55
0.50
β_1
−3.3
−3.0
−2.7
−2.4
β_0

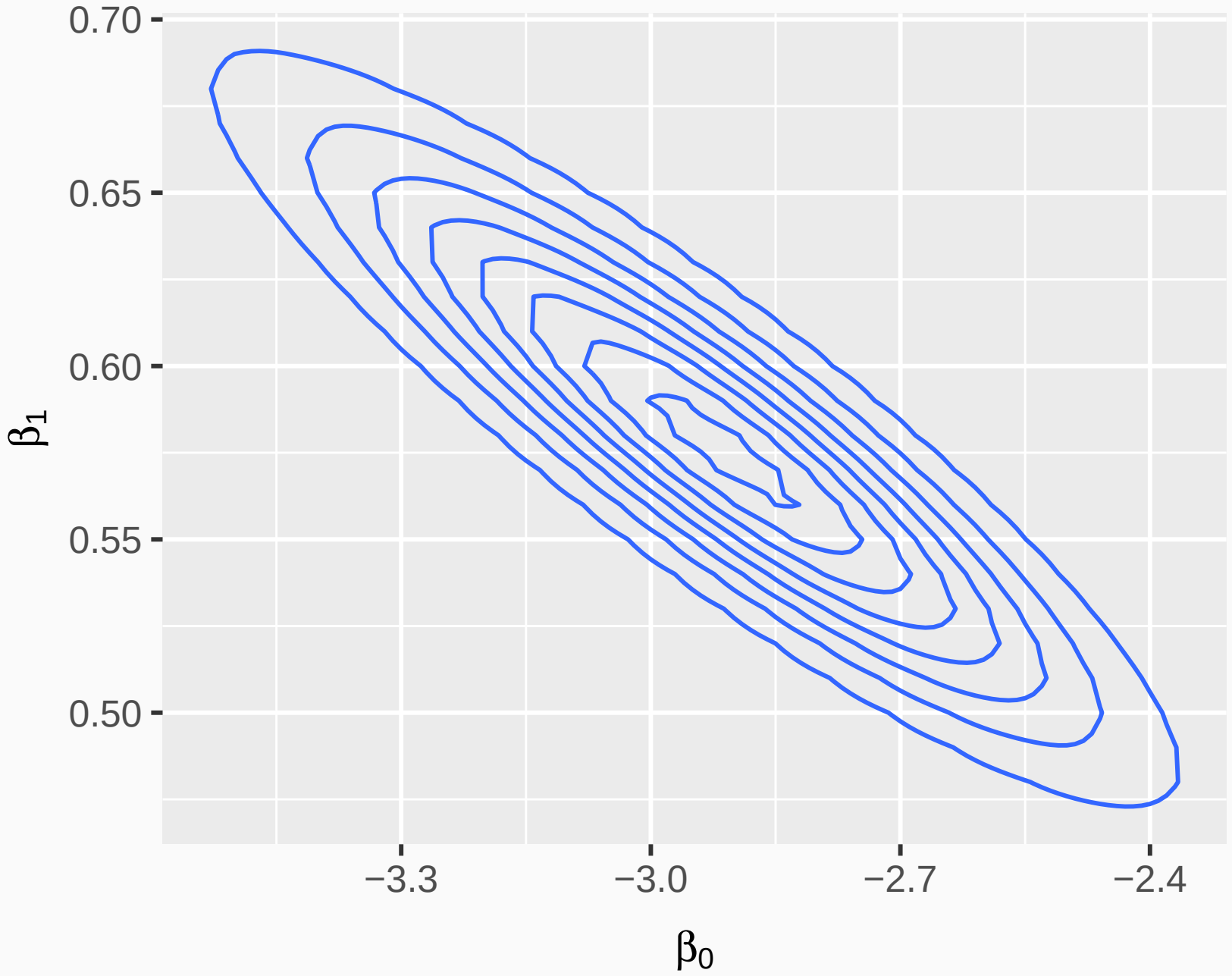
Ajustando o modelo usando a função `optim`

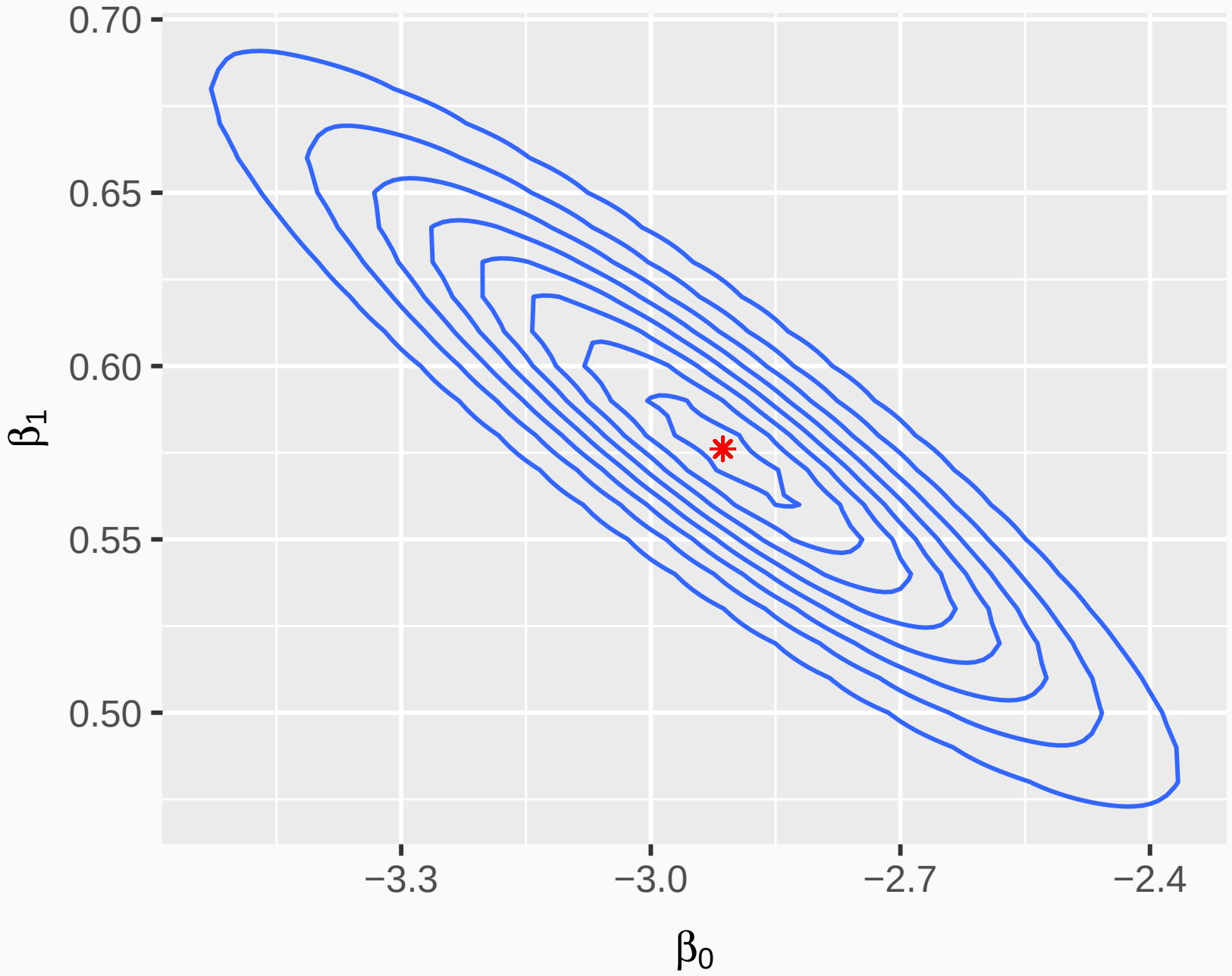
```
aux <- optim(c(0,0), function(.) -log.L(., x=X, y=Y),
             hessian=T)
(beta.hat <- aux$par) # Estimativa dos parâmetros
```

```
## [1] -2.9132931  0.5761001
```

```
solve(aux$hessian)  # Covariâncias
```

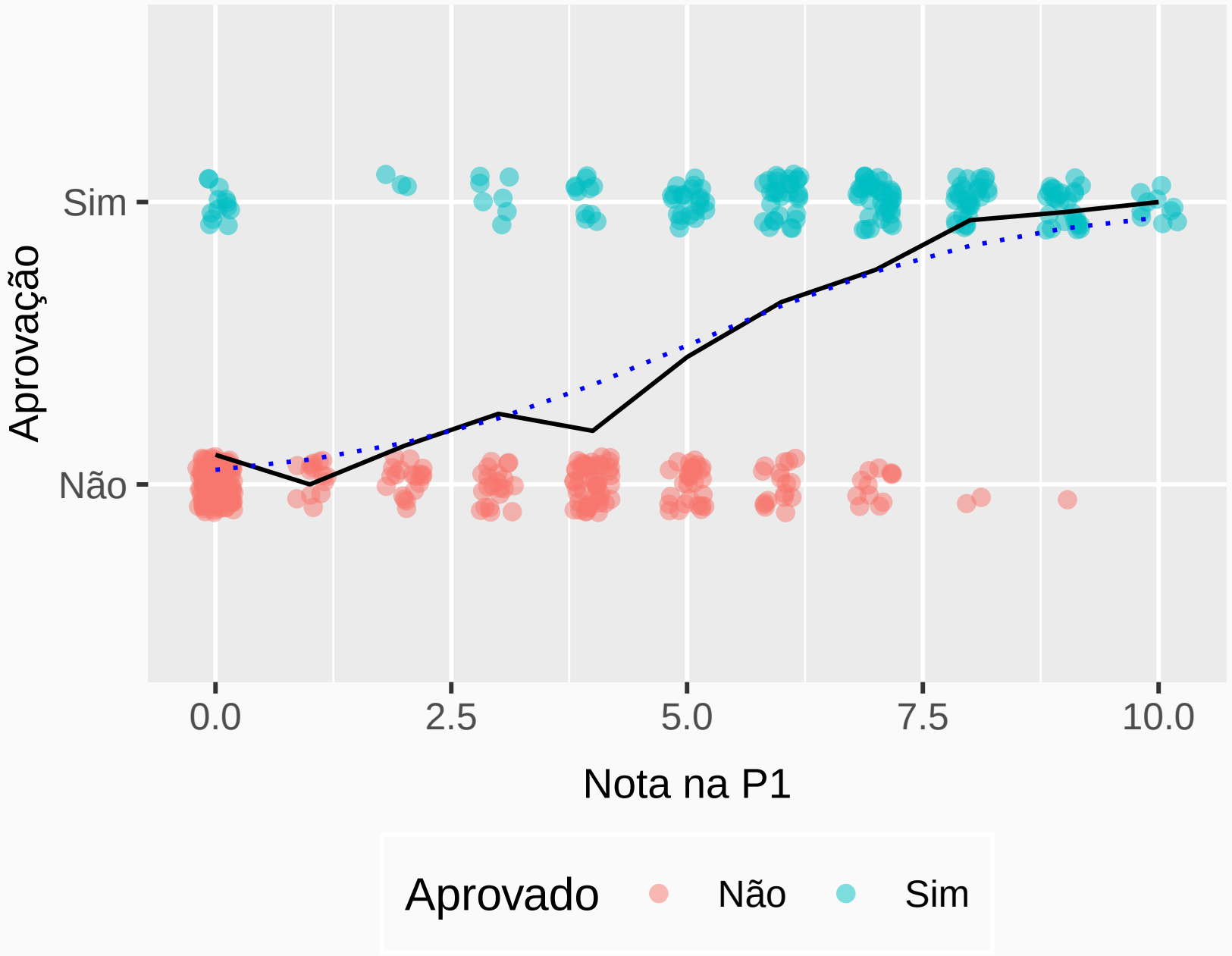
```
##              [,1]         [,2]
## [1,]  0.07896852 -0.013348787
## [2,] -0.01334879  0.002769838
```


0.70
0.65
0.60
0.55
0.50
β_1
−3.3
−3.0
−2.7
−2.4
β_0


0.70
0.65
0.60
0.55
0.50
β1
−3.3
−3.0
−2.7
−2.4
β0

Visualizando a estimativa (dispersão)

```
eta.hat <- cbind(1, 0:10) %*% beta.hat

p.hat <- logistic(eta.hat)

hats <- data.frame(x=0:10, eta.hat, p.hat)

modelo.plot <- grafico.y +

  geom_line(data=hats, aes(x, p.hat), col="blue", lty=3)
```


Aprovação
Sim
Não
0.0
2.5
5.0
7.5
10.0
Nota na P1
Aprovado
Não
Sim