BAYESIAN GENERALIZED LINEAR MODELS

André Santos | andre@metodosexatos.com.br (mailto:andre@metodosexatos.com.br)

09 de janeiro de 2021

Este é um relatório do curso oferecido pela Métodos Exatos (https://www.metodosexatos.com/) feito com o R Notebook.

- Instituição de Ensino: Método Exatos
- Programa: E-learning
- Professor: Ms. André Santos
- Aluno: Seu nome aqui
- RA: '0000000'
- Disciplina: Estatística Bayesiana

1. Estudo de caso:

Os gestores de um e-commerce de moda feminina precisam entender quais os fatores que influenciam as vendas.A amostra é do ano de 2005 e contém informações de 730 clientes de diversas regiões do Brasil.As variáveis de interesse são gênero, frequência de compras no mês, se as pessoas moram na capital, formas de pagamento, quantidade de produtos e valor do pedido.

2. Objetivo geral:

Aplicar um modelo de regressão linear generalizado - Binomial em um relatório de vendas de um e-commerce de moda feminina para inferir sobre propensão à compra.

2.1 Objetivos específicos:

- Inferir sobre quais variáveis têm mais influência nas vendas
- Prever quais clientes têm maior propensão à compra
- Comparar o modelo bayesiano com um modelo matemático de otimização não linear

3. Justificativa na escolha do modelo:

Foi aplicado o modelo binomial sobre os dados pois o objetivo é determinar se os clientes irão ou não comprar no site. Ou seja, nosso problema é binário (compra ou não compra)

4. Dicionário de variáveis

Nome no relatório	Variável	Valor da variável (tipo de dado)
ID cliente	clientes	categórico
Status ID	status	comprou = 1; desistiu = 0
Gênero	genero	homem = 1; mulher = 0
Frequência de compras no mês	compras	discreto (>=0)
Formas de pagamento	pagto	crédito = 1; boleto = 0
Quantidade de itens no pedido	qtde	discreto (>=0)
Valor do pedido com frete	pedido	contínuo (>=0)

5. Amostra

5.1 Dataset original

```
library(readr) # pacote necessário para ler arquivo na web
urlfile="https://raw.githubusercontent.com/metodosexatos/mlgbayes/main/DatasetsES15/ecomm.csv"
mydata<-read_csv2(url(urlfile)) # trocar para "read_csv" para padrão americano
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Parsed with column specification:
## cols(
##   clientes = col_double(),
##   status = col_double(),
##   genero = col_double(),
##   compras = col_double(),
##   regiao = col_double(),
##   pagto = col_double(),
##   qtde = col_double(),
##   pedido = col_double(),
##   amostras = col_character()
## )
```

```
str(mydata) # estrutura da base
```

```
## tibble [730 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ clientes: num [1:730] 1 2 3 4 5 6 7 8 9 10 ...
##  $ status  : num [1:730] 0 0 0 0 0 1 0 1 0 0 ...
##  $ genero  : num [1:730] 0 0 0 1 0 0 0 0 0 1 ...
##  $ compras : num [1:730] 1 1 1 1 1 2 1 2 8 1 ...
##  $ regiao  : num [1:730] 1 0 0 0 0 0 0 0 0 0 ...
##  $ pagto   : num [1:730] 1 1 1 1 1 1 0 1 1 1 ...
##  $ qtde    : num [1:730] 2 1 1 1 5 2 1 2 2 2 ...
##  $ pedido  : num [1:730] 400 1000 100 200 500 400 100 400 500 200 ...
##  $ amostras: chr [1:730] "Treino" "Treino" "Treino" "Teste" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   clientes = col_double(),
##   ..   status = col_double(),
##   ..   genero = col_double(),
##   ..   compras = col_double(),
##   ..   regiao = col_double(),
##   ..   pagto = col_double(),
##   ..   qtde = col_double(),
##   ..   pedido = col_double(),
##   ..   amostras = col_character()
##   .. )
```

5.2 Subset

```
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
```

```
amostra <- mydata %>% filter(amostras == "Treino")
head(amostra)
```

clientes <dbl>	status <dbl>	genero <dbl>	compras <dbl>	regiao <dbl>	pagto <dbl>	qtde <dbl>	pedido <dbl>	amostras <chr>
1	0	0	1	1	1	2	400	Treino
2	0	0	1	0	1	1	1000	Treino
3	0	0	1	0	1	1	100	Treino
15	1	0	1	0	1	2	300	Treino
16	0	0	1	1	1	1	300	Treino
18	1	0	1	0	1	1	300	Treino

6 rows

6. Resumo estatístico da base

6.1 Estatísticas Descritivas

```
summary(amostra[-1])
```

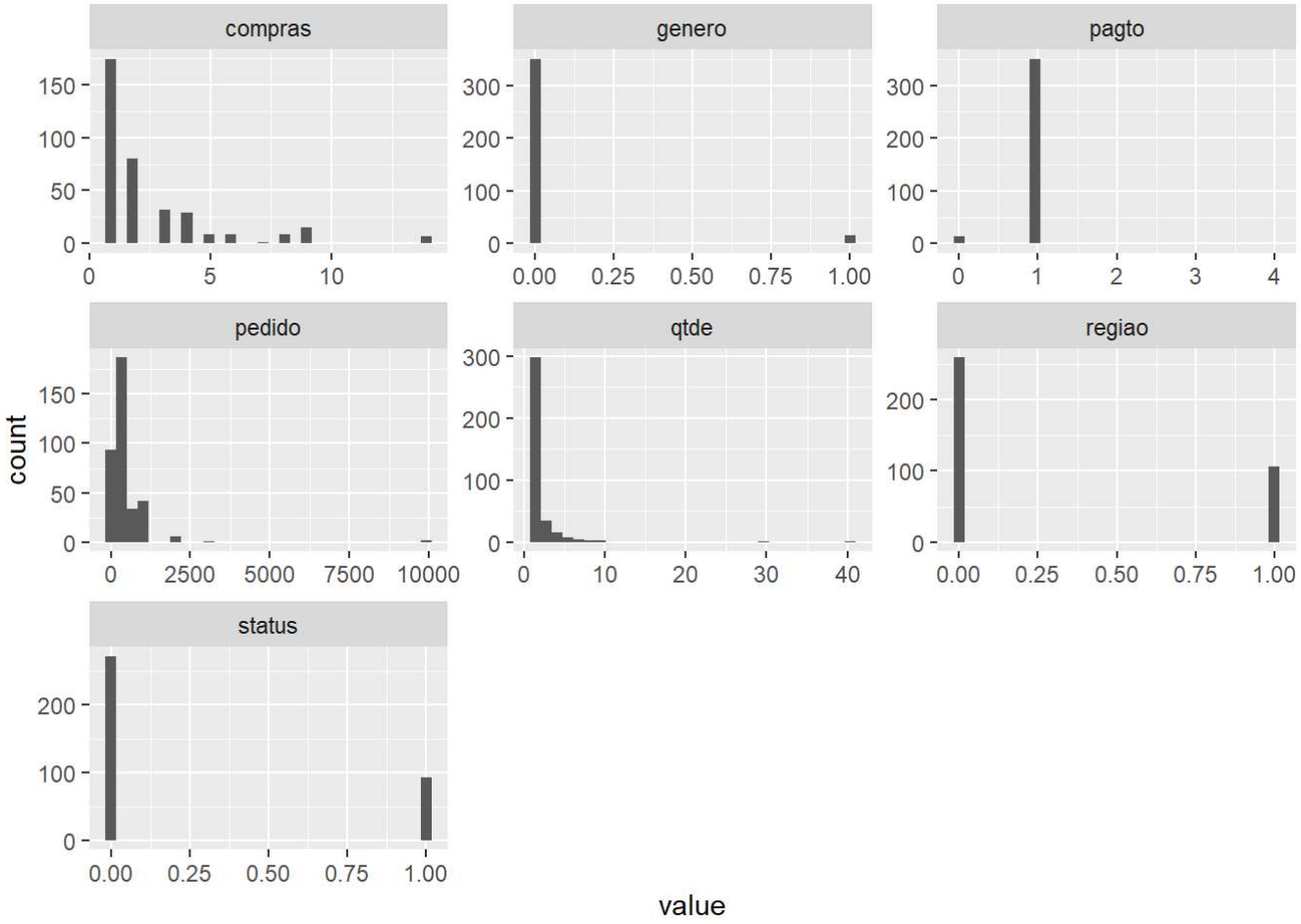
```
##      status           genero          compras           regiao      
##  Min.   :0.0000   Min.   :0.0000   Min.   : 1.000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 1.000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000   Median : 2.000   Median :0.0000  
##  Mean   :0.2548   Mean   :0.0411   Mean   : 2.622   Mean   :0.2904  
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 3.000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :1.0000   Max.   :14.000   Max.   :1.0000  
##      pagto             qtde            pedido          amostras        
##  Min.   :0.0000   Min.   : 1.000   Min.   :   90.0   Length:365        
##  1st Qu.:1.0000   1st Qu.: 1.000   1st Qu.:  100.0   Class :character  
##  Median :1.0000   Median : 1.000   Median :  300.0   Mode  :character  
##  Mean   :0.9699   Mean   : 1.926   Mean   :  455.7                     
##  3rd Qu.:1.0000   3rd Qu.: 2.000   3rd Qu.:  500.0                     
##  Max.   :4.0000   Max.   :40.000   Max.   :10000.0
```

6.2 Histogramas

```
library(purrr)
library(tidyr)
library(ggplot2)
library(dplyr)

histogramas <- select(amostra,status,genero,compras,regiao,pagto,qtde,pedido)
histogramas %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) + facet_wrap(~ key, scales = "free") + geom_histogram()
```

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



7. Bayesian Generalized Linear Model

7.1 Modelo Binomial

```
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.21.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
model_binomial <- stan_glm(status ~ genero+compras+regiao+pagto+qtde+pedido,
                           data = amostra, family = binomial())
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.421 seconds (Warm-up)
## Chain 1:                0.51 seconds (Sampling)
## Chain 1:                0.931 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

```
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.423 seconds (Warm-up)
## Chain 2:                0.521 seconds (Sampling)
## Chain 2:                0.944 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.401 seconds (Warm-up)
## Chain 3:                0.572 seconds (Sampling)
## Chain 3:                0.973 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
```

```
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.413 seconds (Warm-up)
## Chain 4:                0.374 seconds (Sampling)
## Chain 4:                0.787 seconds (Total)
## Chain 4:
```

7.2 Resumo do modelo

```
summary(model_binomial)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      status ~ genero + compras + regiao + pagto + qtde + pedido
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 365
##  predictors:   7
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -1.2    0.6 -1.9  -1.2  -0.5
## genero      -0.9    0.8 -2.0  -0.8   0.1
## compras      0.0    0.0  0.0   0.0   0.1
## regiao      -0.4    0.3 -0.8  -0.4   0.0
## pagto        0.0    0.5 -0.7   0.0   0.6
## qtde         0.1    0.1 -0.1   0.1   0.2
## pedido       0.0    0.0  0.0   0.0   0.0
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.3    0.0  0.2   0.3   0.3
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summa
ry.stanreg')).
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  3631
## genero        0.0  1.0  3540
## compras       0.0  1.0  3460
## regiao        0.0  1.0  3928
## pagto         0.0  1.0  3768
## qtde          0.0  1.0  2269
## pedido        0.0  1.0  2239
## mean_PPD      0.0  1.0  4395
## log-posterior 0.1  1.0  1456
```

```
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, and Rhat is th
e potential scale reduction factor on split chains (at convergence Rhat=1).
```

8. Resultados

8.1 Coeficientes

```
coeff <- exp(model_binomial$coefficients)
coeff
```

```
## (Intercept)      genero     compras      regiao       pagto        qtde
##   0.3097261   0.4491601   1.0424698   0.6638264   0.9796416   1.0658548
##      pedido
##   1.0000154
```

8.2 Interpretação dos coeficientes

```
library(dplyr)
Coeficientes <- c("Intercepto", "genero (beta 1)", "compras (beta 2)", "capital (beta 3)", "pagto (beta 4)", "pedido (beta
 5)",
                  "qtde (beta 6)")
Analise <- c("Dadas todas outras variáveis com valores nulos temos a chance de uma pessoa comprar diminui em 70%",
             "Um cliente ser homem diminui a chance de comprar no site em 43%",
             "Cada pedido a mais feito no site aumenta a chance do cliente comprar em 4%",
             "Pedido feitos por endereços na capital diminui a chance do cliente comprar em 66%",
             "Pedidos feitos com cartão de crédito diminu as chances de realizar a compra em 3%",
             "Cada produto a mais adicionado no pedido aumenta a chance do cliente comprar em 0,06%",
             "O valor do pedido não é significativo para conversão em compra")


result <- cbind(Coeficientes, Analise)
result %>%
  knitr::kable()
```

Coeficientes	Analise

Coeficientes	Analise
Intercepto	Dadas todas outras variáveis com valores nulos temos a chance de uma pessoa comprar diminui em 70%
genero (beta 1)	Um cliente ser homem diminui a chance de comprar no site em 43%
compras (beta 2)	Cada pedido a mais feito no site aumenta a chance do cliente comprar em 4%
capital (beta 3)	Pedido feitos por endereços na capital diminui a chance do cliente comprar em 66%
pagto (beta 4)	Pedidos feitos com cartão de crédito diminu as chances de realizar a compra em 3%
pedido (beta 5)	Cada produto a mais adicionado no pedido aumenta a chance do cliente comprar em 0,06%
qtde (beta 6)	O valor do pedido não é significativo para conversão em compra

9. Considerações

Há uma alta propensão a desistência dos pedidos, sendo os principais fatores de decisão a região e o gênero.

10. Referências

- Como montar tabelas de modelos Bayesianos prontas para publicação (https://storopoli.io/Estatistica-Bayesiana/aux-Tabelas_para_Publicacao.html)
- Estatística Bayesiana com R e Rstan (https://storopoli.io/Estatistica-Bayesiana/)
- GitHub Docs (https://docs.github.com/pt/free-pro-team@latest/github/writing-on-github/organizing-information-with-tables)
- GitHub Métodos Exatos: Bayesian Generalized Linear Model Course (https://github.com/metodosexatos/mlgbayes)
- Kinas, P. 2020. Introdução à Análise Bayesiana (Com R) (https://www.amazon.com.br/Introdu%C3%A7%C3%A3o-%C3%A0-An%C3%A1lise-Bayesiana-Com/dp/6599008828/ref=sr_1_2?dchild=1&keywords=Introdu%C3%A7%C3%A3o+%C3%A0+An%C3%A1lise+Bayesiana+com+R&qid=1609961066&sr=8-2)
- Markdown Guide: Basic Syntx (https://www.markdownguide.org/basic-syntax/) *McElreath,R. 2020.Statistical Rethinking: A Bayesian Course with Examples in R and Stan (https://www.amazon.com.br/Statistical-Rethinking-Bayesian-Course-Examples/dp/1482253445)
- Métodos Exatos: Cursos Avançados para Análise de Dados (https://www.metodosexatos.com/)
- R Markdown: The Definitive Guide (https://bookdown.org/yihui/rmarkdown/)
- rstanarm: Draw from posterior predictive distribution (http://mc-stan.org/rstanarm/reference/posterior_predict.stanreg.html)
- rstanarm: Estimating Generalized Linear Models for Binary and Binomial Data with rstanarm (https://mc-stan.org/rstanarm/articles/binomial.html)

- User-friendly Bayesian regression modeling:A tutorial with `rstanarm` and `shinystan` (http://www.tqmp.org/RegularArticles/vol14-2/p099/p099.pdf)
- Visualizing a Bayesian model (https://s3.amazonaws.com/assets.datacamp.com/production/course_6580/slides/chapter4.pdf)