Introdução a linguagem Stan (*rstan*), um software para modelos bayesianos.



Universidade Federal do Oeste do Pará (UFOPA) Campus de Monte Alegre – Engenharia de Aquicultura

Universidade Federal de Lavras (UFLA) PPG – Estatística e Experimentação Agropecuária

Professor: Carlos Antônio Zarzar

E-mail: carloszarzar_@hotmail.com

carlos.zarzar@ufopa.edu.br

Data: 08/03/2022

AGRADECIMENTO E COLABORADORES:











> Sumário

- Introdução Stan;
 - Sintaxe básica do Stan;
 - Exemplo 1-8 Escolas;
 - Exemplo 2- Regressão linear simples;
 - Exemplo 3 Regressão linear simples com heterocedasticidade; 0
 - Exemplo 4 Modelo de regressão logística;
 - Exemplo 5 Segundo exemplo do modelo de regressão logístico;
 - Exemplo 6 Mais modelos hierárquicos;



carloszarzar_@hotmail.com





> Stan

- Linguagem de programação probabilística para especificar modelos bayesianos estatísticos;
- Stan fornece inferência bayesiana completa por meio de métodos computacional MCMC (Markov Chain Monte Carlo);
- Utiliza-se do algoritmo NUTS (No-U-Turn Sampler), um derivado adaptativo de amostragem Hamiltoniana de Monte Carlo (HMC);
- Stan foi criado por Andrew Gelman e Bob Carpenter, com uma equipe de desenvolvimento composta por 34 membros.
- Stan é nomeado em homenagem a Stanislaw Ulam (1909-1984), co-inventor do método Monte Carlo.
- Ele pode usar MCMC para inferência Bayesiana completa, Variational Bayesian (VB) para inferência Bayesiana aproximada, ou estimativa de máxima verossimilhança (MLE) para estimativa de máxima verossimilhança penalizada;
- Ele define log da função densidade de probabilidade posterior (ou probabilidade penalizada);
- Escrito em C++;









- Stan pode ser acessado por várias interfaces:
 - o CmdStan linha de comando;
 - RStan integrado ao R;
 - PyStan integrado ao Python;
 - MatlabStan integrado ao MATLAB;
 - Stan.il integrado ao Julia;
 - StataStan integrado ao Stata.
- Dentro do R:
 - <u>rstanarm</u>: ajuda o usuário a especificar modelos usando a sintaxe familiar de fórmulas do R (lme4). Já possui os modelos pré-compilado C++.
 - <u>brms</u>: similar ao **rstanarm** pois usa a sintaxe familiar de fórmulas do R, mas dá maior flexibilidade na especificação de modelos mais complexos. Compila os modelos para C++.



➤ Sintaxe básica do Stan

Um modelo Stan é definido por seis blocos de programa:

- ☐ Dados (obrigatório) / data{};
- ☐ Dados transformados / transformed data{};
- ☐ Parâmetros (obrigatório) / parameters{};
- ☐ Parâmetros transformados / transformed parameters{};
- ☐ Modelo (obrigatório) / model{};
- ☐ Quantidades geradas / generated quantities{}.







➤ Sintaxe básica do Stan

Um modelo Stan é definido por seis blocos de programa: ☐ Dados (obrigatório) / data{}; ☐ Dados transformados / transformed data{}; ☐ Parâmetros (obrigatório) / parameters{}; ☐ Parâmetros transformados / transformed parameters{}; ☐ Modelo (obrigatório) / model{}; ☐ Quantidades geradas / generated quantities { }. O bloco de **dados** lê informações externas.

> O bloco de dados transformado permite o préprocessamento dos dados.

```
data {
 int<lower=0> J;
 real y[J];
 real<lower=0> sigma[]];
parameters {
 real mu;
 real<lower=0> tau;
 real theta[];
model {
  // Distribuições a priori
 tau \sim cauchy(0,25);
 theta ~ normal(mu, tau);
   // Verossimilhança
 y ~ normal(theta, sigma);
```

carloszarzar @hotmail.com







➤ Sintaxe básica do Stan

Um modelo Stan é definido por seis blocos de programa: ☐ Dados (obrigatório) / data{};

☐ Dados transformados / transformed data{};

☐ Parâmetros (obrigatório) / parameters{};

☐ Parâmetros transformados / transformed parameters{};

☐ Modelo (obrigatório) / model{};

☐ Quantidades geradas / generated quantities { }.

O bloco de **parâmetros** define o espaço de amostragem

O bloco de **parâmetros transformados** permite o processamento de parâmetros antes que o posterior seja calculado.

```
data {
 int<lower=0> J;
 real y[J];
 real<lower=0> sigma[J];
parameters {
 real mu:
 real<lower=0> tau;
 real theta[];
model {
  // Distribuições a priori
 tau \sim cauchy(0,25);
 theta ~ normal(mu, tau);
   // Verossimilhança
 y ~ normal(theta, sigma);
```







➤ Sintaxe básica do Stan

Um modelo Stan é definido por seis blocos de programa:

- ☐ Dados (obrigatório) / data{};
- ☐ Dados transformados / transformed data{};
- ☐ Parâmetros (obrigatório) / parameters{};
- ☐ Parâmetros transformados / transformed parameters{};
- ☐ Modelo (obrigatório) / model{};
- ☐ Quantidades geradas / generated quantities{}.

No bloco do **modelo**, definimos nosso modelo probabilístico bayesiano.

Por último, o bloco de quantidades geradas permite o pós-processamento.

```
data {
 int<lower=0> J;
 real y[J];
 real<lower=0> sigma[J];
parameters {
 real mu:
 real<lower=0> tau;
 real theta[];
model {
  // Distribuições a priori
 tau \sim cauchy(0,25);
 theta ~ normal(mu, tau);
  // Verossimilhança
 y ~ normal(theta, sigma);
```







➤ Sintaxe básica do Stan

Stan tem dois **tipos** primitivos de variáveis e ambos podem ser limitados.

- int é um tipo inteiro.
- **real** é um tipo de ponto flutuante.

```
int<lower=1> N;
real<upper=5> alpha;
real<lower=-1,upper=1> beta;
real gamma;
real<upper=gamma> zeta;
```

```
As variáveis reais se estendem
  aos tipos de álgebra linear.
```

```
vector[10] a;
                     Column vector
matrix[10, 1] b;
row_vector[10] c; // Row vector
matrix[1, 10] d;
```

Arrays de inteiros, reais, vetores e matrizes estão disponíveis.

```
real a[10];
vector[10] b;
matrix[10, 10] c;
```

carloszarzar @hotmail.com

Stan também implementa uma variedade de variáveis de tipos restritos.

```
simplex[5] theta;
                           sum(theta) = 1
ordered[5] o;
                        // o[1] < ... < o[5]
positive_ordered[5] p;
corr_matrix[5] C;
                        // Symmetric and
                        // positive-definite
cov_matrix[5] Sigma;
```



➤ Sintaxe básica do Stan

Todas as instruções típicas de controle e loop também estão disponíveis.

```
if/then/else
for (i in 1:I)
while (i < I)
```

Existem duas maneiras de declarar as posteriores.

```
~ normal(0, 1);
target += normal_lpdf(y | 0, 1);
# Obsoleto nas novas versões do Stan:
increment_log_posterior(log_normal(y, 0, 1))
```

lpdf – *log probability density function* log da função de densidade de probabilidade da normal de y dado o parâmetro de localização μ e escala σ





➤ Sintaxe básica do Stan

Todas as instruções típicas de controle e loop também estão disponíveis.

```
if/then/else
for (i in 1:I)
while (i < I)
```

E muitas declarações de amostragem são *vetorizadas*.

```
real mu[N];
 real<lower=0> sigma[N];
model {
 // for (n in 1:N)
 // y[n] ~ normal(mu[n], sigma[n]);
 y ~ normal(mu, sigma); // Vectorized version
```

parameters {

Comentário

A posteriori



8 Escolas





Avaliar o programas de *coaching* nos testes de admissão em faculdades





carloszarzar_@hotmail.com





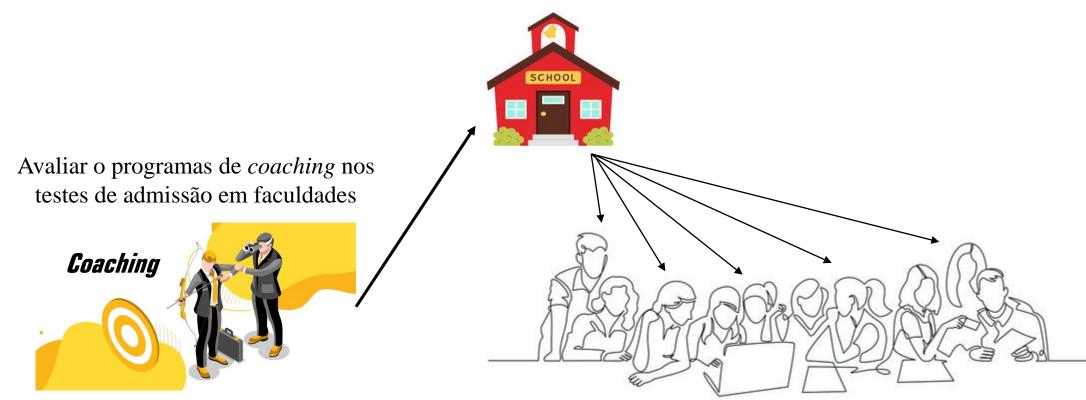
Avaliar o programas de *coaching* nos testes de admissão em faculdades







> Exemplo 8 escolas



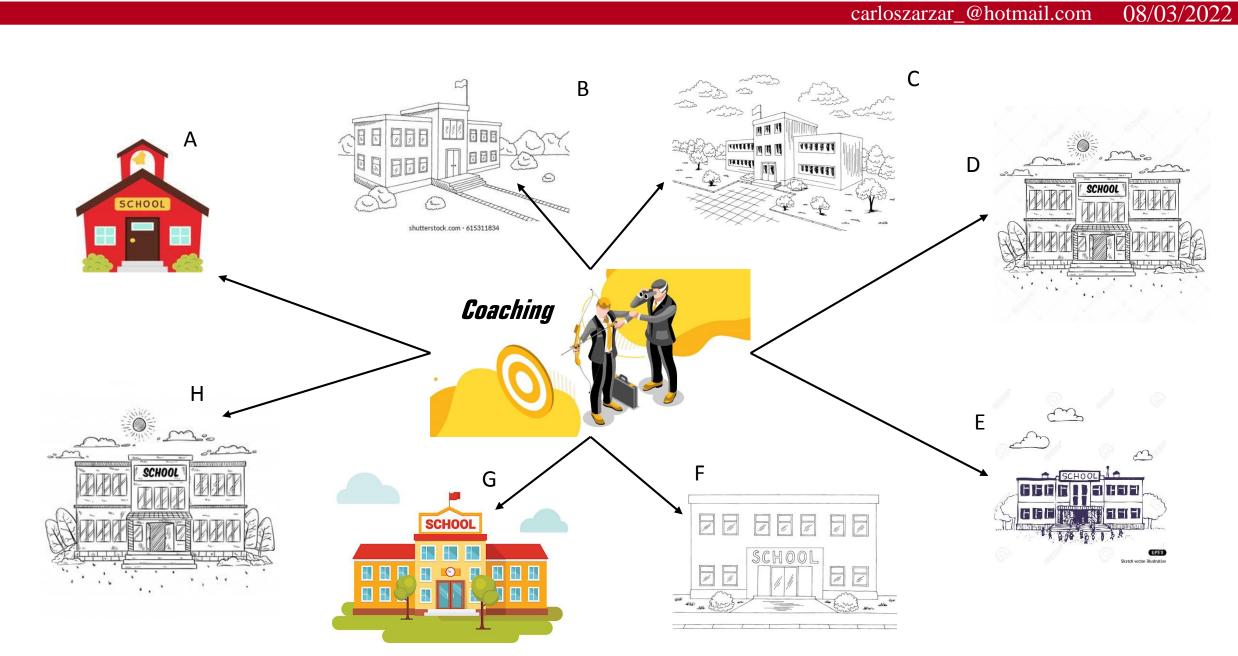
shutterstock.com · 1276275661

$$Score_j \sim N(\mu_j, \sigma_j)$$

 $j = 1, 2, ..., 8$









> Exemplo 8 escolas

Avaliar o programas de *coaching* nos testes de admissão em faculdades



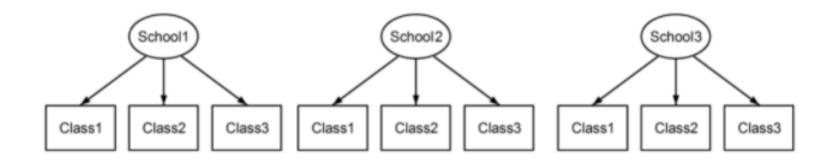
School	Estimate (y_j)	Standard Error (σ_j)
Α	28	15
В	8	10
С	-3	16
D	7	11
E	-1	9
F	1	11
G	18	10
Н	12	18

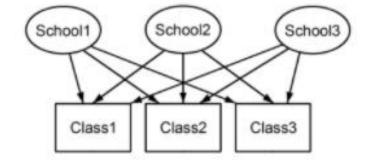
$$Score_j \sim N(\mu_j, \sigma_j)$$

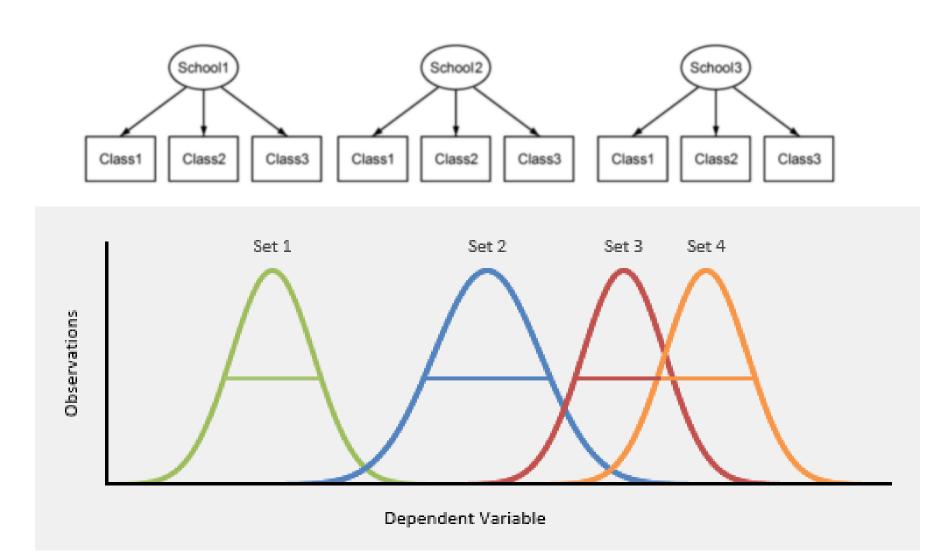
 $j = 1, 2, ..., 8$



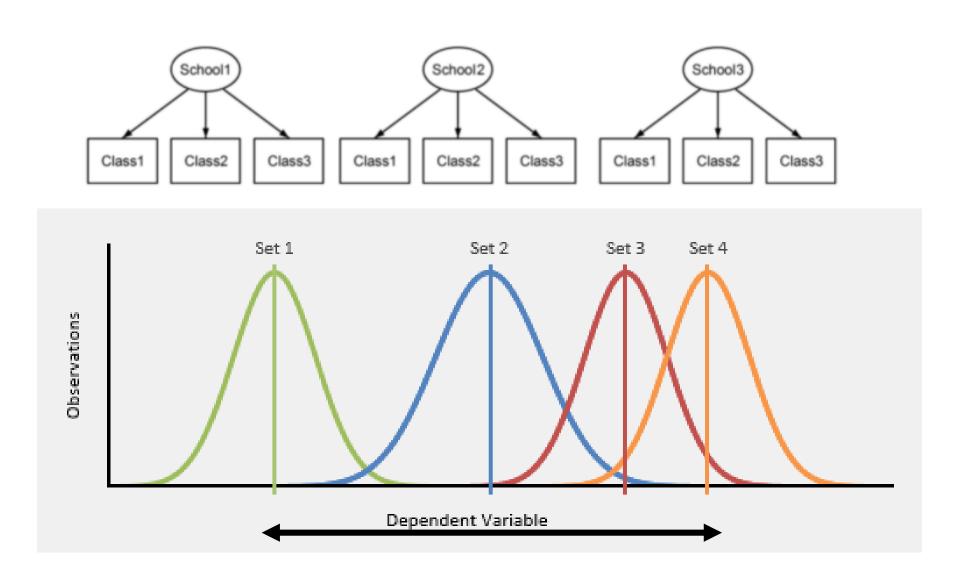
08/03/2022





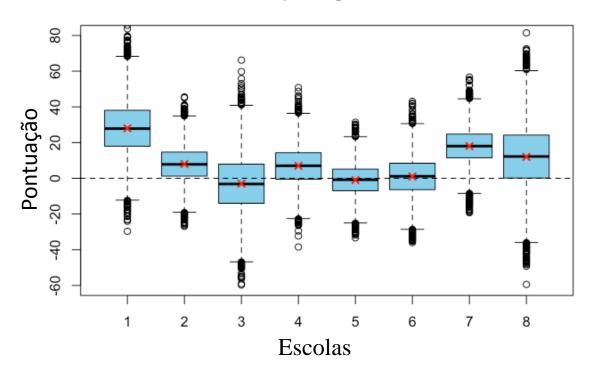


carloszarzar_@hotmail.com 08/03/2022



Não agrupado

No pooling model



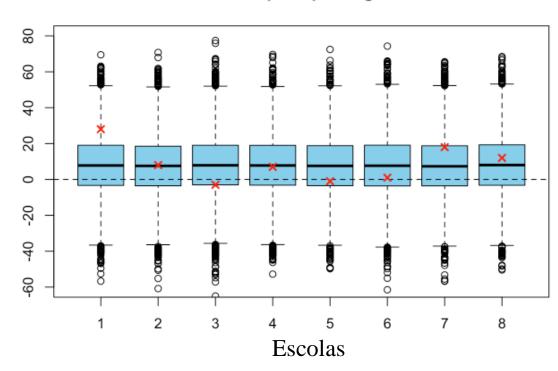
 $Y_j \, | \, heta_j \sim N(heta_j, \sigma_j^2)$ $p(\theta_j) \propto 1$ para todos $j = 1, \ldots, J$.

Priores impróprias não informativas

Completamente agrupado

Complete pooling

carloszarzar_@hotmail.com



$$Y_j \, | \, heta \sim N(heta, \sigma_j^2) \quad ext{para todos } j = 1, \ldots, J$$
 $p(heta) \propto 1$

Priores impróprias não informativas





> Exemplo 8 escolas

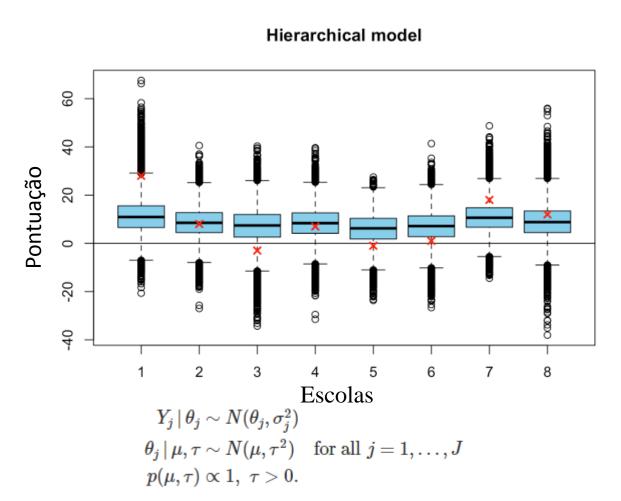
Priori imprópria

Uma priori $\pi(\boldsymbol{\theta})$ é dita ser uma **priori imprópria** se

$$\int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\pi(\boldsymbol{\theta})\mathrm{d}\;\boldsymbol{\theta}=\infty.$$

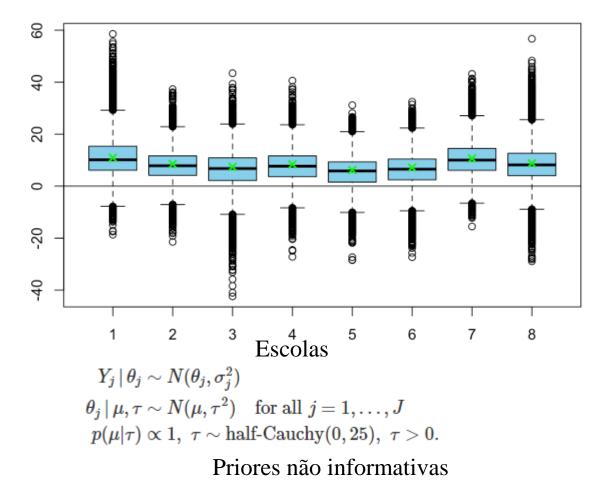
Em diversas situações, apesar da priori ser imprópria, a posteriori poderá ser própria. No entanto, nestas condições é necessário verificar.

O uso de prioris impróprias é interessante para se definir prioris não informativas. Um exemplo de tal A priori de Jeffreys é usada para se definir prioris não informativas.



Priores impróprias não informativas

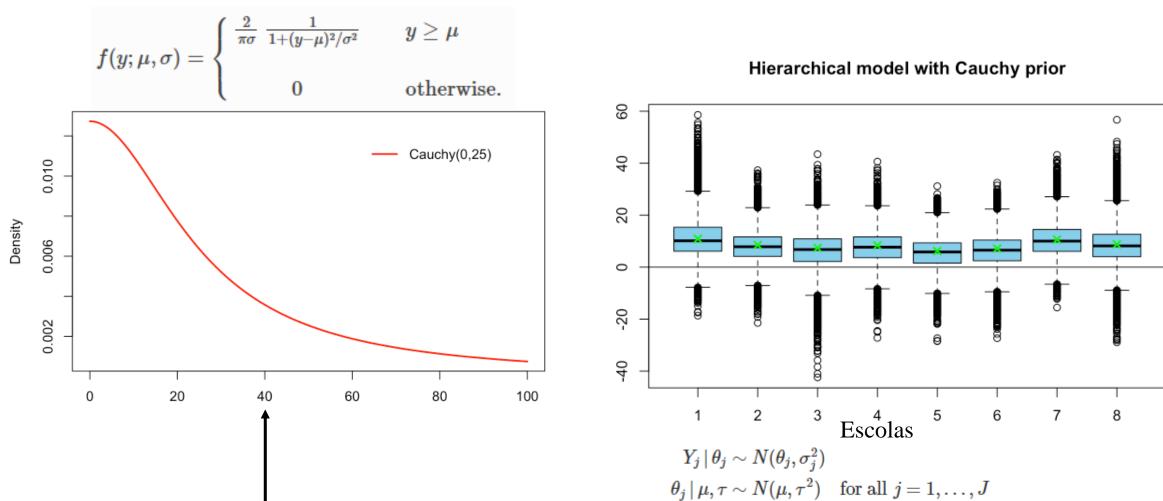
Hierarchical model with Cauchy prior







Função de densidade de probabilidade



Priores não informativas

 $p(\mu| au) \propto 1, \,\, au \sim ext{half-Cauchy}(0,25), \,\, au > 0.$

00000



carloszarzar_@hotmail.com 08/03/20

```
data {
  int<lower=0> J;
  real y[J];
  real<lower=0> sigma[]];
parameters {
  real mu;
  real<lower=0> tau;
  real theta[];
model {
   // Distribuições a priori
  tau \sim cauchy(0,25);
  theta ~ normal(mu, tau);
   // Verossimilhança
  y ~ normal(theta, sigma);
```





> Exemplo 8 escolas

Resultados

```
Inference for Stan model: 8School.
4 chains, each with iter=10000; warmup=5000; thin=2;
post-warmup draws per chain=2500, total post-warmup draws=10000.
                              2.5%
                                             50%
                                      25%
                                                    75% 97.5% n_eff Rhat
                          sd
          mean se_mean
                                            7.86
          7.86
                  0.10 5.02
                             -2.03
                                     4.63
                                                  11.16 17.61
                                                               2577 1.00
mu
          6.27
                  0.14 4.72
                              0.79
                                     2.73
                                            5.15
                                                   8.60 18.07
                                                               1097 1.01
tau
theta[1]
         10.99
                  0.15 7.98 -2.40
                                     5.72
                                           10.12 15.11 29.87
                                                               2887 1.00
          7.88
                                     3.85
                                            7.83 11.86 20.31 3858 1.00
theta[2]
                  0.10 6.25 -4.44
theta[3]
          6.23
                  0.12 7.60 -10.99
                                     2.02
                                            6.68
                                                  10.99 20.24
                                                               4287 1.00
          7.63
                                                  11.68 20.49
theta[4]
                  0.10 6.47 -5.31
                                     3.61
                                            7.69
                                                               4328 1.00
theta[5]
          5.17
                  0.12 6.26 -7.92
                                     1.28
                                            5.53
                                                   9.40 16.56
                                                               2853 1.00
theta[6]
          6.14
                  0.10 6.57 -7.81
                                     2.20
                                            6.44
                                                  10.47 18.46
                                                               4133 1.00
                                     5.96
                                            9.93 14.42 25.74 3153 1.00
theta[7]
         10.54
                  0.12 6.73 -1.14
theta[8]
          8.24
                  0.11 7.63 -6.76
                                     3.69
                                            8.14 12.56 24.24
                                                               4448 1.00
1p__
         -17.21
                  0.28 5.81 -27.43 -21.36 -17.79 -13.48 -4.29
                                                                419 1.01
Samples were drawn using NUTS(diag_e) at Wed Mar 02 10:05:41 2022.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```





> Exemplo 8 escolas

Configuração da amostragem MCMC

```
Inference for Stan model: 8School.
4 chains, each with iter=10000; warmup=5000; thin=2;
post-warmup draws per chain=2500, total post-warmup draws=10000.
                                       25%
                                              50%
                                                     75% 97.5% n_eff Rhat
                               2.5%
                          sd
           mean se_mean
                                             7.86
           7.86
                   0.10 5.02
                              -2.03
                                      4.63
                                                   11.16 17.61
                                                                2577 1.00
mu
           6.27
                   0.14 4.72
                               0.79
                                      2.73
                                             5.15
                                                    8.60 18.07
                                                                1097 1.01
tau
theta[1]
          10.99
                   0.15 7.98 -2.40
                                      5.72
                                            10.12
                                                   15.11 29.87
                                                                2887 1.00
           7.88
                                      3.85
                                                  11.86 20.31 3858 1.00
theta[2]
                   0.10 6.25 -4.44
                                             7.83
                                                                4287 1.00
theta[3]
           6.23
                   0.12 7.60 -10.99
                                      2.02
                                             6.68
                                                   10.99 20.24
           7.63
                                                   11.68 20.49
theta[4]
                   0.10 6.47 -5.31
                                      3.61
                                             7.69
                                                                4328 1.00
theta[5]
           5.17
                   0.12 6.26 -7.92
                                      1.28
                                             5.53
                                                    9.40 16.56
                                                                2853 1.00
theta[6]
           6.14
                   0.10 6.57 -7.81
                                      2.20
                                             6.44
                                                   10.47 18.46
                                                                4133 1.00
          10.54
                                      5.96
                                                               3153 1.00
theta[7]
                   0.12 6.73 -1.14
                                             9.93
                                                  14.42 25.74
theta[8]
           8.24
                   0.11 7.63 -6.76
                                      3.69
                                             8.14 12.56 24.24
                                                                4448 1.00
1p__
         -17.21
                   0.28 5.81 -27.43 -21.36 -17.79 -13.48 -4.29
                                                                 419 1.01
Samples were drawn using NUTS(diag_e) at Wed Mar 02 10:05:41 2022.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

0000







Estimativa dos parâmetros

```
Inference for Stan model: 8School.
4 chains, each with iter=10000; warmup=5000; thin=2;
post-warmup draws per chain=2500, total post-warmup draws=10000.
                                              50%
                                                     75% 97.5% n_eff Rhat
                               2.5%
                                       25%
                          sd
           mean se_mean
                              -2.03
                                             7.86
                                                   11.16 17.61
                                                               2577 1.00
           7.86
                   0.10 5.02
                                      4.63
mu
           6.27
                   0.14 4.72
                               0.79
                                      2.73
                                             5.15
                                                    8.60 18.07
                                                               1097 1.01
tau
theta[1]
          10.99
                   0.15 7.98 -2.40
                                      5.72
                                            10.12
                                                   15.11 29.87
                                                                2887 1.00
           7.88
                                      3.85
                                             7.83 11.86 20.31
                                                               3858 1.00
theta[2]
                   0.10 6.25 -4.44
                   0.12 7.60 -10.99
theta[3]
           6.23
                                      2.02
                                             6.68
                                                   10.99 20.24
                                                                4287 1.00
           7.63
                                                   11.68 20.49
                                                                4328 1.00
theta[4]
                   0.10 6.47 -5.31
                                      3.61
                                             7.69
theta[5]
           5.17
                   0.12 6.26 -7.92
                                      1.28
                                             5.53
                                                    9.40 16.56
                                                                2853 1.00
theta[6]
           6.14
                   0.10 6.57 -7.81
                                      2.20
                                             6.44
                                                   10.47 18.46
                                                                4133 1.00
          10.54
                                      5.96
                                             9.93 14.42 25.74
                                                                3153 1.00
theta[7]
                   0.12 6.73 -1.14
                   0.11 \ 7.63 \ -6.76
theta[8]
           8.24
                                      3.69
                                             8.14 12.56 24.24
                                                                4448 1.00
         -17.21
                   0.28 5.81 -27.43 -21.36 -17.79 -13.48 -4.29
                                                                 419 1.01
1p___
Samples were drawn using NUTS(diag_e) at Wed Mar 02 10:05:41 2022.
```

0000

For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).

Inferência Bayesiana





> Exemplo 8 escolas

Inference for Stan model: 8School. 4 chains, each with iter=10000; warmup=5000; thin=2; post-warmup draws per chain=2500, total post-warmup draws=10000. 2.5% 25% 50% 75% 97.5% n_eff Rhat sd mean se_mean 7.86 7.86 0.10 5.02 -2.034.63 11.16 17.61 2577 1.00 mu 8.60 18.07 6.27 0.14 4.72 0.79 2.73 5.15 1097 1.01 tau theta[1] 10.99 0.15 7.98 -2.40 5.72 10.12 15.11 29.87 2887 1.00 7.88 3.85 3858 1.00 theta[2] 0.10 6.25 -4.44 7.83 11.86 20.31 0.12 7.60 - 10.994287 1.00 theta[3] 6.23 2.02 6.68 10.99 20.24 7.63 11.68 20.49 4328 1.00 theta[4] 0.10 6.47 -5.31 3.61 7.69 9.40 16.56 2853 1.00 theta[5] 5.17 0.12 6.26 -7.92 1.28 5.53 theta[6] 6.14 0.10 6.57 -7.81 2.20 6.44 10.47 18.46 4133 1.00 10.54 5.96 3153 1.00 theta[7] 0.12 6.73 -1.14 9.93 14.42 25.74 theta[8] 8 24 0 11 7 63 -6 76 3 69 8 14 12 56 24 24 4448 1.00 419 1.01 -17.21 0.28 5.81 -27.43 -21.36 -17.79 -13.48 -4.29 1p_

00000

Samples were drawn using NUTS(diag_e) at Wed Mar 02 10:05:41 2022. For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).

Densidade de probabilidade a posterior logarítmizada não normalizada.

Posteriori não normalizada:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

Posteriori normalizada:

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

Posteriori normalizada log:

$$\log p(\theta|D) = \log(p(D|\theta)) + \log(p(\theta))$$



Estatísticas de diagnóstico sobre a convergência da cadeia e o número de sorteios independentes na amostra.

```
Inference for Stan model: 8School.
4 chains, each with iter=10000; warmup=5000; thin=2;
post-warmup draws per chain=2500, total post-warmup draws=10000.
                                       25%
                                               50%
                                                      75% 97.5% n_eff Rhat
                               2.5%
                          sd
           mean se_mean
                                              7.86
                                                    11.16 17.61
           7.86
                   0.10 5.02
                              -2.03
                                      4.63
                                                                25// 1.00
mu
                                                     8.60 18.07
           6.27
                   0.14 4.72
                               0.79
                                      2.73
                                              5.15
                                                                 1097 1.01
tau
theta[1]
          10.99
                   0.15 7.98
                             -2.40
                                      5.72
                                             10.12
                                                    15.11 29.87
                                                                 2887 1.00
           7.88
                                      3.85
                                                                 3858 1.00
theta[2]
                   0.10 6.25
                             -4.44
                                             7.83
                                                    11.86 20.31
theta[3]
           6.23
                   0.12 7.60 -10.99
                                      2.02
                                              6.68
                                                    10.99 20.24
                                                                 4287 1.00
           7.63
                                                    11.68 20.49
                                                                 4328 1.00
theta[4]
                   0.10 6.47 -5.31
                                      3.61
                                              7.69
                                                                 2853 1.00
theta[5]
           5.17
                   0.12 6.26
                              -7.92
                                      1.28
                                              5.53
                                                     9.40 16.56
                                                                 4133 1.00
theta[6]
           6.14
                   0.10 6.57 -7.81
                                      2.20
                                              6.44
                                                    10.47 18.46
          10.54
                   0.12 6.73 -1.14
                                      5.96
                                                                 3153 1.00
theta[7]
                                              9.93
                                                    14.42 25.74
theta[8]
           8.24
                   0.11 7.63
                             -6.76
                                      3.69
                                              8.14
                                                   12.56 24.24
                                                                 4448 1.00
         -17.21
                   0.28 5.81 -27.43 -21.36 -17.79 -13.48 -4.29
                                                                  419 1.01
1p__
```

Samples were drawn using NUTS(diag_e) at Wed Mar 02 10:05:41 2022. For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).

Tamanho efetivo da amostra (n_eff)

Estatística de redução de escala potencial Rhat

Gelman e Rubin: Rhat > 1.05 indica cadeia não convergiu





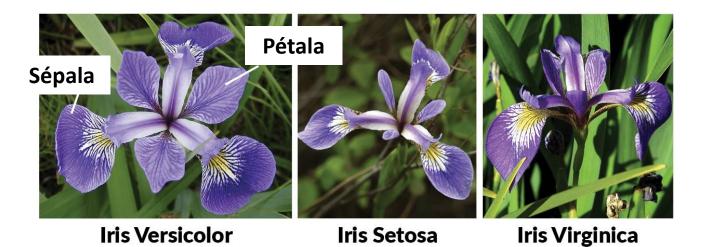
Regressão linear simples





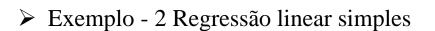
> Exemplo - 2 Regressão linear simples

```
# Dados
## http://www.lac.inpe.br/
data("iris")
head(iris)
```



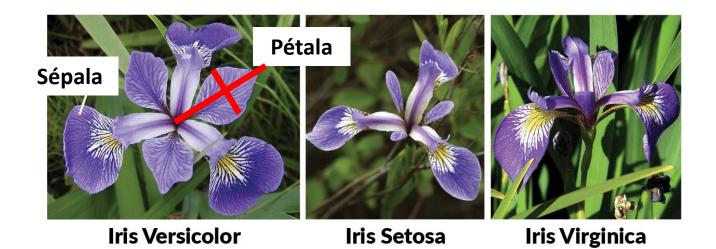


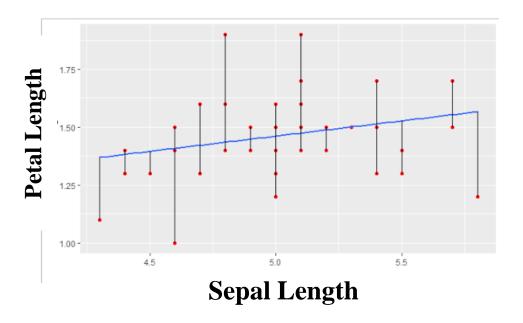




```
Dados
   http://www.lac.inpe.br
data("iris")
head(iris)
```

Exemplo 2











Exemplo - 2 Regressão linear simples





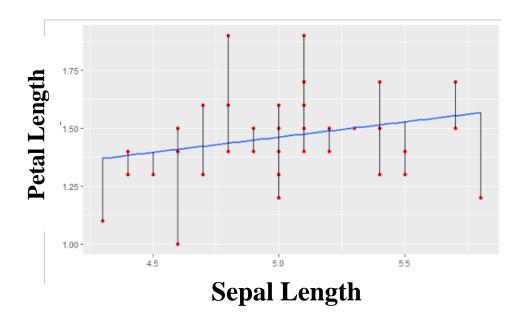


Iris Setosa



carloszarzar_@hotmail.com

Iris Virginica



- > Dados:
 - Número de obs.;
 - $Y-Petal\ length;$
 - X Sepal length;

- Parâmetros:
 - Alpha a;
 - Beta β;
 - Sigma σ ;

➤ Modelo:

$$y_i \sim N(\alpha + x_i \cdot \beta, \sigma)$$

Priores impróprias não informativas Uniform($-\infty$; $+\infty$)

Informação a priori





Exemplo - 2 Regressão linear simples



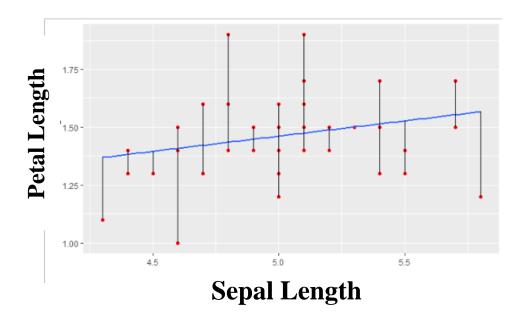




Iris Setosa



Iris Virginica



- > Dados:
 - Número de obs.;
 - $Y-Petal\ length;$
 - X Sepal length;

- Parâmetros:
 - Alpha a;
 - Beta β;
 - Sigma σ;

$$y_i \sim N(\alpha \ + \ x_i \ \cdot \ \beta \ , \ \sigma)$$

$$\alpha \sim N(10 \ , \ 0.1)$$
 Informação a priori



Exemplo - 2 Regressão linear simples

0000

```
Stan model for simple linear regression
data {
int < lower = 1 > N; // Sample size
vector[N] x; // Predictor
vector[N] y; // Outcome
parameters {
real alpha; // Intercept
real beta; // Slope (regression coefficients)
real < lower = 0 > sigma; // Error SD
model {
// alpha ~ normal(10, 0.1);
// beta ~ normal(1, 0.1);
y ~ normal(alpha + x * beta , sigma);
generated quantities {
real y_rep[N];
for (n in 1:N) {
y_{rep}[n] = normal_{rng}(x[n] * beta + alpha, sigma);
```

Random number generation distributions



Exemplo - 2 Regressão linear simples

Resultados

Função lm()

0000

```
Call:
lm(formula = Petal.Length ~ Sepal.Length, data = dat)
Coefficients:
 (Intercept) Sepal.Length
     0.8031
                    0.1316
```

Função RStan

```
Inference for Stan model: stan_model1.
4 chains, each with iter=1000; warmup=500; thin=1;
post-warmup draws per chain=500, total post-warmup draws=2000.
                               25%
                                     50%
                                           75% 97.5% n_eff Rhat
      mean se_mean
                     sd 2.5%
alpha 0.80
              0.02 0.36 0.10
                              0.57
                                    0.80
                                         1.02 1.52
                                                       504 1.01
beta
      0.13 0.00 0.07 -0.01 0.09
                                    0.13 0.18 0.27
                                                      502 1.01
              0.00 0.02 0.14 0.16 0.17 0.18 0.21
sigma 0.17
                                                       879 1.00
     61.61
              0.05 1.19 58.58 61.06 61.91 62.48 62.99
                                                       643 1.00
1p__
Samples were drawn using NUTS(diag_e) at Fri Mar 04 18:23:09 2022.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

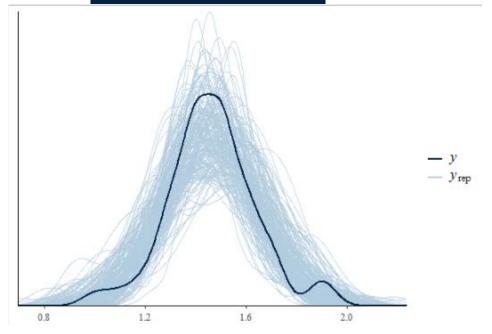




> Exemplo - 2 Regressão linear simples

Função Rstan Priores não informativas

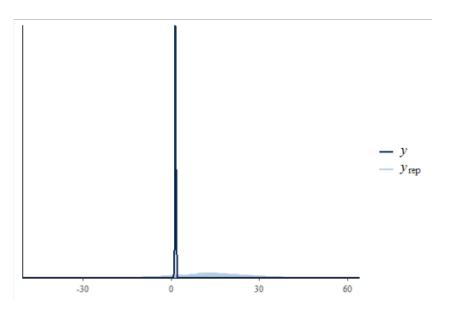
	mean	se_mean
alpha	0.81	0.01
beta	0.13	0.00
sigma	0.17	0.00
1p	61.53	0.04



Resultados

Função Rstan Priores muito informativas

	mean	se_mean
alpha	9.96	0.00
beta	0.80	0.00
sigma	12.82	0.03
1p	-152.36	0.03









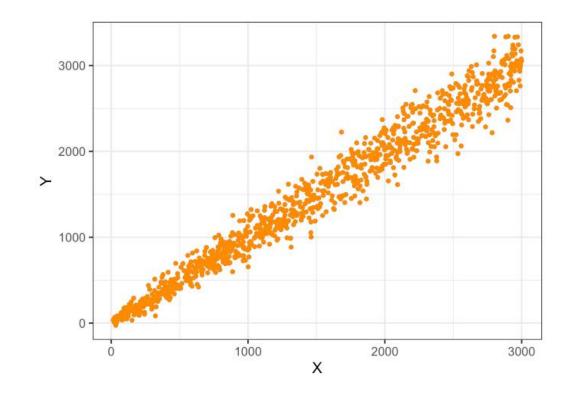




Regressão linear simples com heterocedasticidade



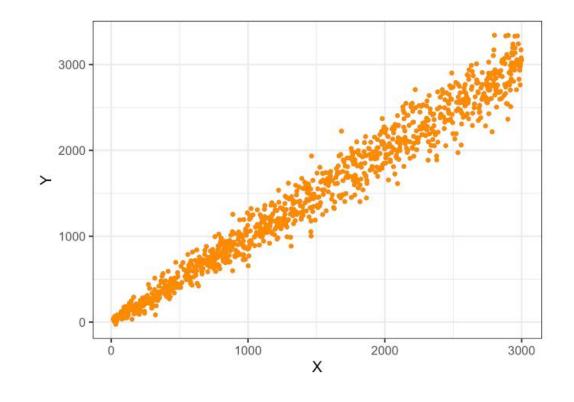
> Exemplo - 3 Regressão linear simples com heterocedasticidade



$$Y_i = \beta x_i + \epsilon_i, \ i = 1, \dots n$$

$$\epsilon_i \sim N(0,\sigma_i) ext{ independentes, com } \sigma_i = \alpha \sqrt{|x_i|},$$

> Exemplo - 3 Regressão linear simples com heterocedasticidade

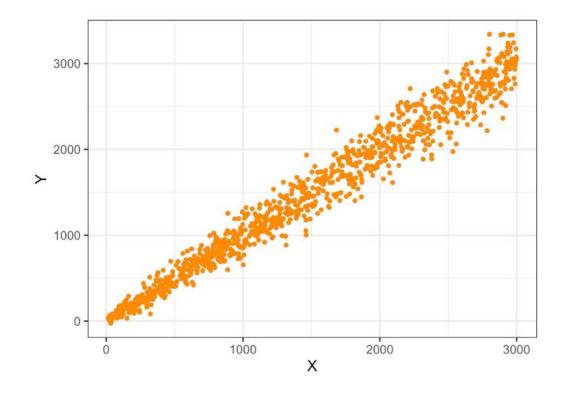


$$Y_i = \beta x_i + \epsilon_i, \; i=1,\dots n$$
 $\epsilon_i \sim N(0,\sigma_i) ext{ independentes, com } \sigma_i = \sqrt{|x_i|},$ $eta = 1$ e $lpha = 4$.





Exemplo - 3 Regressão linear simples com heterocedasticidade



$$Y_i=eta x_i+\epsilon_i,\ i=1,\dots n$$
 $\epsilon_i\sim N(0,\sigma_i)$ independentes, com $\sigma_i=lpha\sqrt{|x_i|},$ $eta=1$ e $lpha=4.$

```
rm(list = ls())
# pacotes
library(ggplot2)
# Tamanho da amostra
N < -1000
# fixando uma semente (número pseudoaleatório)
set.seed(11071995)
X \leftarrow sample((N/100):(N*3), N)
Y \leftarrow rnorm(N,1*X,4*sqrt(X))
qplot(X,Y) +
  theme_bw(15) +
  geom_point(color = 'darkorange')
```

Exemplo - 3 Regressão linear simples com heterocedasticidade

```
data {
 int<lower=0> N;
 vector[N] y;
 vector[N] x;
parameters {
 real beta;
 real<lower=0> alpha;
model {
 beta \sim normal(0,10);
 alpha \sim gamma(1,1);
 y ~ normal(beta * x, alpha * sqrt(x));
```

- > Dados:
 - N Número de obs.;
 - *Y Variável dependente*;
 - *X Variável independente*;
- ➤ Modelo:

$$Y_i = \beta x_i + \epsilon_i, \ i = 1, \dots n$$

$$\epsilon_i \sim N(0, \sigma_i)$$
 independentes, com $\sigma_i = \alpha \sqrt{|x_i|}$, $p(\beta) \sim N(0, 10)$ $p(\alpha) \sim Gamma(1, 1)$

- > Parâmetros:
 - Alpha a;
 - Beta β;





Regressão linear simples com heterocedasticidade Resultados

$$Y_i=eta x_i+\epsilon_i,\ i=1,\dots n$$
 $\epsilon_i\sim N(0,\sigma_i)$ independentes, com $\sigma_i=lpha\sqrt{|x_i|},$ $eta=1$ e $lpha=4$.

```
Inference for Stan model: 01faf712f58ce4399a6e006770d47aff.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
                               2.5%
                                         25%
                                                  50%
                                                           75%
                                                                  97.5% n_eff Rhat
         mean_se_mean
                        sd
                               0.99
                                        1.00
                                                 1.00
                                                          1.00
                                                                   1.00 2243
beta
          1.0
                 0.00 0.00
alpha
                 0.00 0.09
                               3.92
                                        4.04
                                                 4.10
                                                          4.17
                                                                   4.29
                                                                        1647
          4.1
                 0.03 1.03 -5438.34 -5435.89 -5435.19 -5434.79 -5434.52 1292
      -5435.5
Samples were drawn using NUTS(diag_e) at Fri Mar 04 18:39:10 2022.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```





Modelo de regressão logística





Exemplo - 4 Modelo de regressão logística

- Regressão logística é usada para estimar probabilidade de um resultado binário;
- É um caso especial dos GLMs;
- Ela se baseia na relação entre a razão de chances e probabilidades;

Linear~ Losgística

ODDs ~ *Probability*

Distribuição de Bernoulli:

(Aprovado | Reprovado);

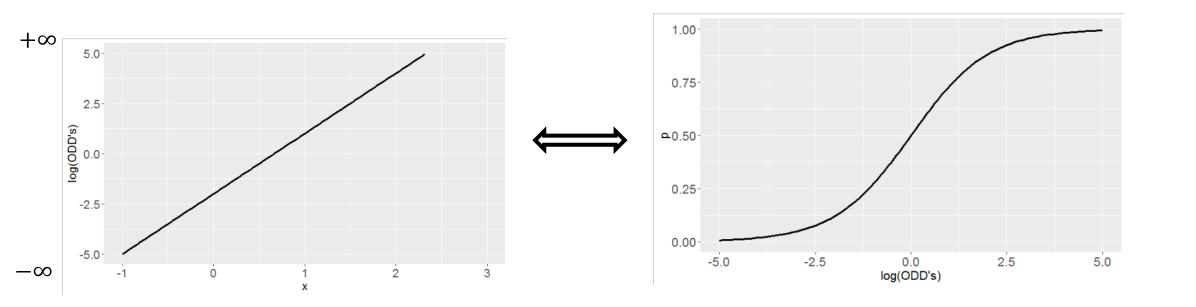
(Doente | Não Doente);

(Venceu | Perdeu);

(Verdadeiro | Falso);

(Sim | Não);

(1 | 0);







> Exemplo - 4 Modelo de regressão logística

- Regressão logística é usada para estimar probabilidade de um resultado binário;
- É um caso especial dos GLMs;
- Ela se baseia na relação entre a razão de chances e probabilidades;

Linear~ Losgística

ODDs ~ *Probability*

Distribuição de Bernoulli:

(Aprovado | Reprovado);

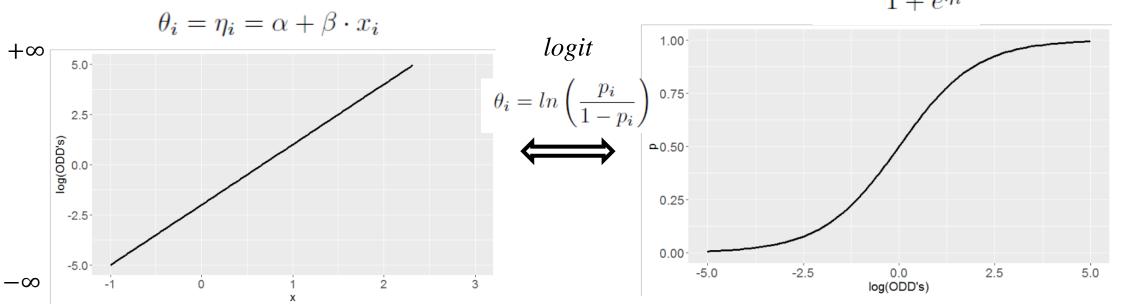
(Doente | Não Doente);

(Sim | Não);

(1 | 0);

$$f(y_i \mid p_i) = p_i^{y_i} (1 - p_i)^{1 - y_i} \; ; \; y_i = 0, 1$$

$$p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}$$







Exemplo - 4 Modelo de regressão logística

- Regressão logística é usada para estimar probabilidade de um resultado binário;
- É um caso especial dos GLMs;
- Ela se baseia na relação entre a razão de chances e probabilidades;

Linear~ Losgística

ODDs ~ *Probability*

- Considere dados de inspeções na detecção de rachaduras para estudos do estado das estruturas (ciência dos matérias);
- Detecção (dados simulados):
 - 1- Há rachadura (profundidade mm da rachadura na estrutura);
 - 0- Não há dano detectado na estrutura local;

Distribuição de Bernoulli:

carloszarzar_@hotmail.com

(Aprovado | Reprovado); (Doente | Não Doente); (Venceu | Perdeu); (Verdadeiro | Falso); (Sim | Não); (1 | 0);







- > Exemplo 4 Modelo de regressão logística
 - > Dados:
 - N Número de obs.;
 - *det* Detecção fissura 1,0;
 - *depth* Profundidade mm;
 - K Número de predições;
 - depth_pred valores da profundidade da fissura a serem previstas;







- > Exemplo 4 Modelo de regressão logística
 - > Dados:
 - N Número de obs.;
 - *det* Detecção fissura 1,0;
 - *depth* Profundidade mm;
 - K Número de predições;
 - depth_pred valores da profundidade da fissura a serem previstas;

➤ Modelo:

Distribuição de Bernoulli: bernoulli_logit(η)

$$f(y_i \mid p_i) = p_i^{y_i} (1 - p_i)^{1 - y_i} ; y_i = 0, 1$$

Modelo $logit^{-1}$:

$$p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}$$

$$\theta_i = \eta_i = \alpha + \beta \cdot x_i$$

$$p(\beta) \sim N(1, 0.25)$$

$$p(\alpha) \sim N(-2.5, 0.5)$$

1 - Detectou; 0 - Não detectou;





- > Exemplo 4 Modelo de regressão logística
 - > Dados:
 - N Número de obs.;
 - *det* Detecção fissura 1,0;
 - *depth* Profundidade mm;
 - K Número de predições;
 - depth_pred valores da profundidade da fissura a serem previstas;

➤ Modelo:

Distribuição de Bernoulli: bernoulli $_{logit}(\eta)$

$$f(y_i \mid p_i) = p_i^{y_i} (1 - p_i)^{1 - y_i} \; ; \; y_i = 0, 1$$

Modelo $logit^{-1}$:

$$p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}} \longrightarrow g^{-1}: logit$$
Profundio

$$\theta_i = \eta_i = \alpha + \beta \cdot x_i$$
 Profundidade depth

$$g^{-1}$$
: logit







- Exemplo 4 Modelo de regressão logística
 - > Dados:
 - N Número de obs.;
 - *det* Detecção fissura 1,0;
 - *depth* Profundidade mm;
 - K Número de predições;
 - depth_pred valores da profundidade da fissura a serem previstas;
 - Parâmetros:
 - Alpha α;
 - Beta β;

➤ Modelo:

Distribuição de Bernoulli: bernoulli_logit(η)

$$f(y_i \mid p_i) = p_i^{y_i} (1 - p_i)^{1 - y_i} \; ; \; y_i = 0, 1$$

Modelo $logit^{-1}$:

$$p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}$$

$$\theta_i = \eta_i = 0 + \beta x_i$$

$$p(\beta) \sim N(1, 0.25)$$

$$p(\alpha) \sim N(-2.5, 0.5)$$





Exemplo - 4 Modelo de regressão logística

- > Dados:
 - N Número de obs.;
 - *det* Detecção fissura 1,0;
 - *depth* Profundidade mm;
 - K Número de predições;
 - depth_pred valores da profundidade da fissura a serem previstas;
- Parâmetros:
 - Alpha α;
 - Beta β;
- Quantidades Geradas:
 - postpred_pr Probabilidade a posteriori das predições;
 - Inv_logit(η) função inversa logística aplicada a x;

➤ Modelo:

Distribuição de Bernoulli: bernoulli_logit(η)

$$f(y_i \mid p_i) = p_i^{y_i} (1 - p_i)^{1 - y_i} \; ; \; y_i = 0, 1$$

Modelo $logit^{-1}$:

$$p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}$$

$$\theta_i = \eta_i = \alpha + \beta \cdot x_i$$

$$p(\beta) \sim N(1, 0.25)$$

$$p(\alpha) \sim N(-2.5, 0.5)$$





Exemplo - 4 Modelo de regressão logística

```
int <lower = 0> N; // Defining the number of defects in the test dataset
 int <lower = 0, upper = 1> det [N]; // A variable that describes whether each defect was detected [1]or not [0]
 vector <lower = 0> [N] depth; // A variable that describes the corresponding depth of each defect
 int <lower = 0> K; // Defining the number of probabilistic predictions required from the model
 vector <lower = 0> [K] depth_pred;
parameters {
 real alpha;
 real beta;
model {
 det ~ bernoulli_logit(alpha + beta * depth);
 alpha \sim normal(-2.5, 0.5);
 beta ~ normal(1, 0.25);
generated quantities {
 // Using the fitted model for probabilistic prediction.
  // K posterior predictive distributions will be estimated for a corresponding crack depth
 vector [K] postpred_pr;
  for (k in 1:K) {
    postpred_pr[k] = inv_logit(alpha + beta * depth_pred[k]);
```





Exemplo - 4 Modelo de regressão logística

Resultados

Valores reais

```
Dados simulados
## dados das inspeções para
set.seed(1008)
  Parâmetros para simulação
N < -500
lower <- 0
upper <- 10
alpha_true <- -2.5
beta_true <- 1.3
```

carloszarzar @hotmail.com

Função GLM

```
Coefficients:
            Estimate Std. Error
             -2.7912
(Intercept)
                         0.3693
depth
              1.4303
                         0.1564
```

Função RStan

```
Inference for Stan model: 62104baa107fde848271122633daf7a7.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
        mean se_mean
                       sd 2.5%
                                      25%
                                              50%
                                                      75%
                                                            97.5% n_eff Rhat
alpha
        -2.56
                0.01 0.27
                            -3.11 -2.74
                                            -2.55
                                                    -2.37
                                                            -2.05 1305
        1.32
                0.00 0.11
                            1.11
                                     1.25
                                             1.32
                                                     1.40
                                                            1.55 1350
beta
                0.03 1.01 -111.51 -109.14 -108.43 -108.04 -107.77
1p___
      -108.76
                                                                   960
Samples were drawn using NUTS(diag_e) at Fri Mar 04 18:00:31 2022.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```







Segundo exemplo do modelo de regressão logístico



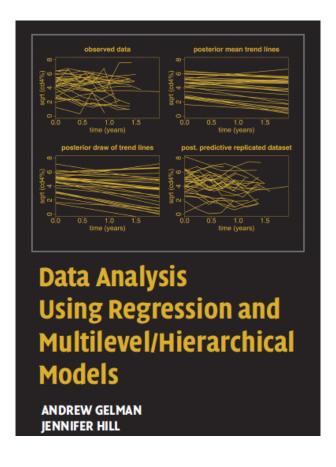




- > Exemplo 5 Segundo exemplo do modelo de regressão logístico
 - Dados:
- Obtido no livro de Andrew Gelman e Jennifer Hill;

Exemplo 5





carloszarzar_@hotmail.com



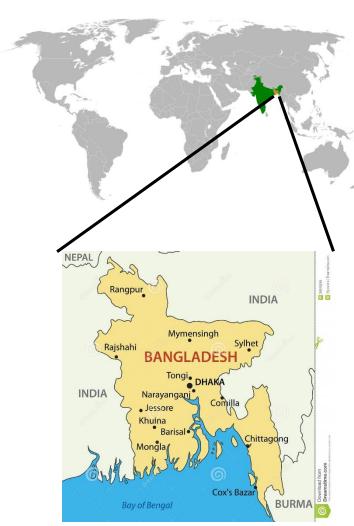


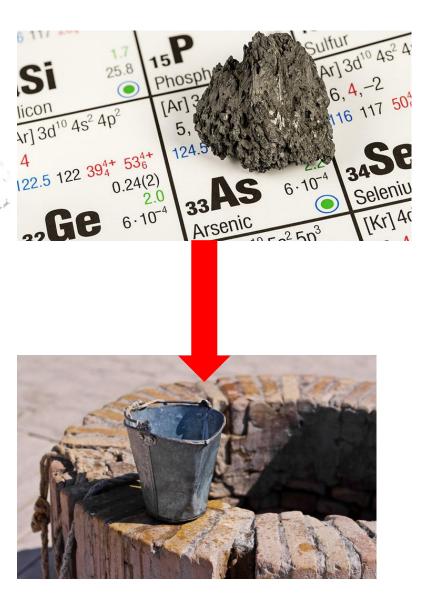
Exemplo - 5 Segundo exemplo do modelo de regressão logístico

- Dados:
- Obtido no livro de Andrew Gelman e Jennifer Hill;

Exemplo 5

Sobre contaminação dos poços artesanais na região de Bangladesh;











> Exemplo - 5 Segundo exemplo do modelo de regressão logístico

> Dados:

Devido a contaminação por arsênio na região de Bangladesh foi perguntado em uma entrevista com os moradores se eles estavam dispostos a obter água potável de outro poço.

- Vaiável dependente:
- 1 Sim interessado;
- 0 Não interessado;





carloszarzar_@hotmail.com



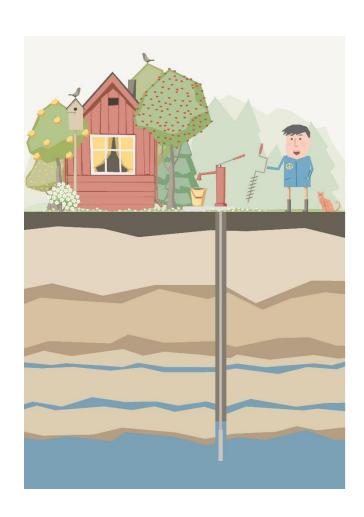


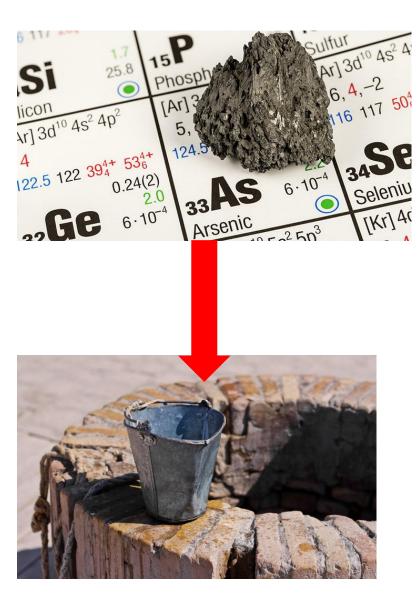
Exemplo - 5 Segundo exemplo do modelo de regressão logístico

> Dados:

Devido a contaminação por arsênio na região de Bangladesh foi perguntado em uma entrevista com os moradores se eles estavam dispostos a obter água potável de outro poço.

- Vaiável dependente:
- 1 Sim interessado;
- 0 Não interessado;
- Vaiável independente: Informações sobre a família;
- Nível de arsênio na água dos moradores;
- Distância da casa até o poço seguro mais próximo;









Exemplo - 5 Segundo exemplo do modelo de regressão logístico

> Dados:

Devido a contaminação por arsênio na região de Bangladesh foi perguntado em uma entrevista com os moradores se eles estavam dispostos a obter água potável de outro poço.

- Vaiável dependente:
- 1 Sim interessado;
- 0 Não interessado;
- Vaiável independente: Informações sobre a família;
- Nível de arsênio na água dos moradores;
- Distância da casa até o poço seguro mais próximo;

- > Dados:
 - N int, Número de obs.;
 - P-int, Número de preditores;
 - a real, Priori parâmetro t-students (graus de lib.);

carloszarzar @hotmail.com

- X **Matriz** de delineamento;
- Y **Vetor int** binário (0 ou 1);
- > Parâmetros:
 - beta0 real;
 - beta vector[P];
- ➤ Modelo: **T**-students $p(\beta_0) \sim T(\nu = 7 \; , \; \mu = a \; , \; \sigma = 0.1)$ $p(\beta) \sim T(\nu = 7, \mu = 0, \sigma = 1)$ $Y \sim Bernoulli_logit(\beta_0 + X\beta)$





Exemplo - 5 Segundo exemplo do modelo de regressão logístico

Exemplo 5

```
data
  int<lower=0> N; // Número de obs;
  int<lower=0> P; // Número de preditores (2)
  real a; // parametro da priori t-students (graus de liberdade)
  matrix[N,P] x; // Matrix delineamento
  int<lower=0,upper=1> y[N]; // preditor
parameters{
 real beta0;
  vector[P] beta;
model{
 beta0 ~ student_t(7,a,0.1);
 beta ~ student_t(7,0,1);
  y ~ bernoulli_logit(beta0+x*beta);
```

> Dados:

- N int, Número de obs.:
- P **int**, Número de preditores;
- a real, Priori parâmetro t-students (graus de lib.);

carloszarzar @hotmail.com

- X **Matriz** de delineamento;
- Y **Vetor int** binário (0 ou 1);
- > Parâmetros:
 - beta0 **real**:
 - beta vector[P];

Modelo: T-students
$$p(\beta_0) \sim T(\nu = 7 \; , \; \mu = a \; , \; \sigma = 0.1)$$

$$p(\beta) \sim T(\nu = 7 \; , \mu = 0 \; , \; \sigma = 1)$$

$$Y \sim Bernoulli_logit(\beta_0 + X\beta)$$





Exemplo - 5 Segundo exemplo do modelo de regressão logístico

Resultados

1 - Sim interessado;

Exemplo 5

Distância da casa até o poço Nível de arsênio na água

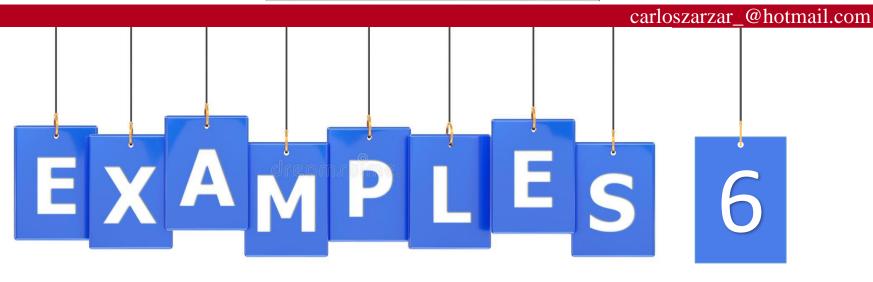
```
Inference for Stan model: bc39cac0b4d697301a99db383651f9dc.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
```

```
sd 2.5%
                               25%
                                      50%
                                            75% 97.5% n_eff Rhat
        mean se_mean
        0.33
                  0 0.04 0.26 0.31 0.33 0.35 0.40
beta0
                                                      3180
beta[1] -0.34
                  0 0.04 -0.42 -0.37 -0.35 -0.32 -0.27
                                                      3554
beta[2] 0.51
                  0 0.05 0.42 0.48 0.51 0.54 0.60 3328
```

Samples were drawn using NUTS(diag_e) at Fri Mar 04 18:08:59 2022. For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).







Mais modelos hierárquicos





- > Exemplo 6 Mais modelos hierárquicos
 - Dados "sleepstudy" no pacote "lme4";
 - o Reaction: Tempo médio de reação (ms);
 - Days: Número de dias de privação de sono;
 - Subject: Código (ID) do participante;



O estudo queria quantificar o efeito da privação do sono no tempo de reação das pessoas submetido ao teste;

Belenky et al. (2003)





Exemplo - 6 Mais modelos hierárquicos

- Dados "sleepstudy" no pacote "lme4";
 - o Reaction (y): Tempo médio de reação (ms);
 - Days (x): Número de dias de privação de sono;
 - Subject (level): Código (ID) do participante;

$$y_{ij} = \beta_0 + u_{0j} + (\beta_1 + u_{1j}) \cdot \text{Days} + e_i$$

$$egin{bmatrix} u_0 \ u_1 \end{bmatrix} \sim \mathcal{N}\left(egin{bmatrix} 0 \ 0 \end{bmatrix}, \Omega = egin{bmatrix} \sigma_0^2 & \operatorname{cov}\left(u_0, u_1
ight) \ \operatorname{cov}\left(u_0, u_1
ight) & \sigma_1^2 \end{bmatrix}
ight)$$



O estudo queria quantificar o efeito da privação do sono no tempo de reação das pessoas submetido ao teste;

Belenky et al. (2003)

$$e \sim N(0, \sigma_e)$$

Queremos estimar a matriz de correlação completa dos efeitos aleatórios







- > Exemplo 6 Mais modelos hierárquicos
- > Dados:
 - N int, Número de obs.;
 - Days **int**, Número de dias de privação de sono (0 a 9);
 - *RT* **real**, Reaction Time;
 - J **int,** Número de grupos;
 - Subject **Vetor int** binário (0 a J);





- Exemplo 6 Mais modelos hierárquicos
- > Dados:
 - N int, Número de obs.;
 - Days **int**, Número de dias de privação de sono (0 a 9);
 - RT **real**, Reaction Time;
 - J int, Número de grupos;
 - Subject Vetor int binário (0 a J);
- > Parâmetros:
 - beta **vector**[2];
 - sigma_e **real<lower=0>** desvio padrão (DP) do residual;
 - sigma_u **vector**<**lower**=**0**>[**2**] DP efeito aleatório;
 - L_u cholesky_fator_corr[2] fator de Cholesky (de uma matriz de correlação 2x2);
 - z_u **matrix[2,J]** matriz de efeito aleatório;

A matriz de covariância é parametrizada pela matriz de correlação com o fator de Cholesky.

carloszarzar @hotmail.com





Exemplo - 6 Mais modelos hierárquicos

🌓 🧶 🌑 Exemplo 6

- > Dados:
 - N int, Número de obs.;
 - Days **int**, Número de dias de privação de sono (0 a 9);
 - *RT* **real**, Reaction Time;
 - J int, Número de grupos;
 - Subject **Vetor int** binário (0 a J);
- > Parâmetros:
 - beta **vector**[2];
 - sigma_e **real<lower=0>** desvio padrão (DP) do residual;
 - sigma_u **vector**<**lower=0**>[2] DP efeito aleatório;
 - L_u cholesky_fator_corr[2] fator de Cholesky (de uma matriz de correlação 2x2);
 - z_u **matrix[2,J]** matriz de efeito aleatório;
- > Transformed parameters:
 - u matrix[2,J], é matriz de efeito aleatório transformada; u = diag_pre_multiply(sigma_u, L_u) * z_u; -

Diagonal resultante da multiplicação do vetor sigma_u %*% L_u pela matrix E depois multiplicado pela matriz de efeito aleatório z u.





- Exemplo 6 Mais modelos hierárquicos
- > Dados:
 - N int, Número de obs.;
 - Days **int**, Número de dias de privação de sono (0 a 9);
 - RT **real**, Reaction Time;
 - J int, Número de grupos;
 - Subject **Vetor int** binário (0 a J);
- > Parâmetros:
 - beta **vector**[2];
 - sigma_e real<lower=0> desvio padrão (DP) do residual;
 - sigma_u **vector**<**lower**=**0**>[**2**] DP efeito aleatório;
 - L_u cholesky_fator_corr[2] fator de Cholesky (de uma matriz de correlação 2x2);
 - z_u **matrix[2,J]** matriz de efeito aleatório;
- > Transformed parameters:
 - u matrix[2,J], é matriz de efeito aleatório transformada; u = diag_pre_multiply(sigma_u, L_u) * z_u;

Modelo:

$$y_{ij} = \beta_0 + u_{0j} + (\beta_1 + u_{1j}) \cdot \text{Days} + e_i$$

$$egin{bmatrix} u_0 \ u_1 \end{bmatrix} \sim \mathcal{N}\left(egin{bmatrix} 0 \ 0 \end{bmatrix}, \Omega = egin{bmatrix} \sigma_0^2 & \operatorname{cov}\left(u_0, u_1
ight) \ \operatorname{cov}\left(u_0, u_1
ight) & \sigma_1^2 \end{bmatrix}
ight)$$

$$e \sim N(0, \sigma_e)$$

Distribuição a priori:

$$p(\sigma_e) \sim N(0, 5)$$

 $p(\beta_0) \sim N(0.3, 0.5)$
 $p(\beta_1) \sim N(0.2, 2)$
 $p(Z_u) \sim N(0, 2)$
 $p(L_u) \sim LKJ(1.5)$

Distribuição Lewandowski-Kurowicka-Joe (LKJ)

Distribuição de probabilidade para matrizes de correlação positiva definida, ou equivalentemente para seus fatores de Cholesky





Exemplo - 6 Mais modelos hierárquicos

Resultados

$$y_{ij} = eta_0 + u_{0j} + (eta_1 + u_{1j}) \cdot \mathrm{Days} + e_i \ egin{bmatrix} u_0 \ u_1 \end{bmatrix} \sim \mathcal{N}\left(egin{bmatrix} 0 \ 0 \end{bmatrix}, \Omega = egin{bmatrix} \sigma_0^2 & \mathrm{cov}\left(u_0, u_1
ight) \ \mathrm{cov}\left(u_0, u_1
ight) \end{bmatrix}
ight)$$

carloszarzar @hotmail.com

 $e \sim N(0, \sigma_e)$ Inference for Stan model: sleep_model. 4 chains, each with iter=2000; warmup=1000; thin=1; post-warmup draws per chain=1000, total post-warmup draws=4000.

```
mean se_mean
                          sd 2.5% 97.5% n_eff Rhat
beta[1]
          0.251
                     0 0.008 0.236 0.266 2342 1.001
          0.010
beta[2]
                     0 0.002 0.007 0.014 2667 1.000
sigma_u[1] 0.014
                     0 0.004 0.008 0.022 2004 1.000
sigma_u[2] 0.003
                     0 0.001 0.002 0.005 1749 1.001
          0.026
                     0 0.002 0.023 0.029 3419 1.000
sigma_e
```

Samples were drawn using NUTS(diag_e) at Fri Mar 04 18:19:04 2022. For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).

Efeito fixo

Desvios padrão do efeitos aleatórios

Erro residual



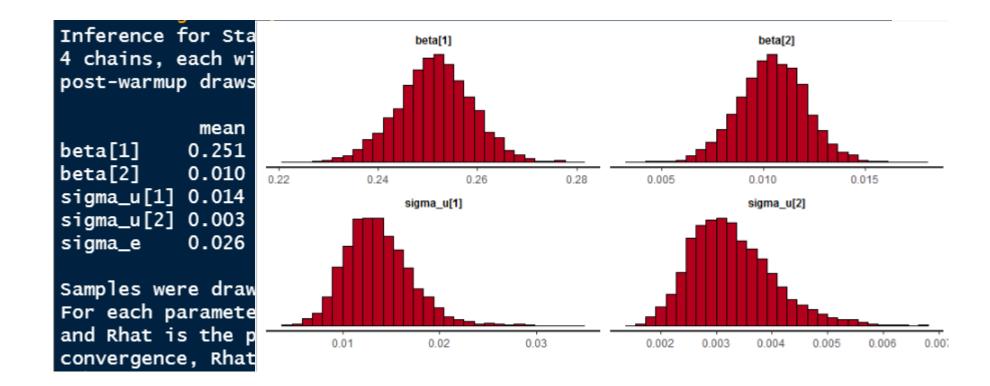




> Exemplo - 6 Mais modelos hierárquicos

● ● ● ● Exemplo 6

Resultados







Exemplo - 6 Mais modelos hierárquicos

Resultados

$$y_{ij} = eta_0 + u_{0j} + (eta_1 + u_{1j}) \cdot \mathrm{Days} + e_i \ egin{bmatrix} u_0 \ u_1 \end{bmatrix} \sim \mathcal{N}\left(egin{bmatrix} 0 \ 0 \end{bmatrix}, \Omega = egin{bmatrix} \sigma_0^2 & \mathrm{cov}\left(u_0, u_1
ight) \ \mathrm{cov}\left(u_0, u_1
ight) \end{bmatrix}
ight)$$

carloszarzar @hotmail.com

Matriz de correlação

$$e \sim N(0, \sigma_e)$$

```
Inference for Stan model: sleep_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
                                       25%
                                             50%
                           sd 2.5%
                                                  75% 97.5% n_eff Rhat
           mean se_mean
Omega[1,1] 1.000
                    Nan 0.000 1.00 1.000 1.000 1.000
                                                              NaN
                                                                    NaN
Omega[1,2] 0.079 0.008 0.292 -0.49 -0.125 0.078 0.289 0.637 1243 1.002
Omega[2,1] 0.079 0.008 0.292 -0.49 -0.125 0.078 0.289 0.637 1243 1.002
                  0.000 0.000 1.00 1.000 1.000 1.000 1.000 3676 0.999
Omega[2,2] 1.000
Samples were drawn using NUTS(diag_e) at Fri Mar 04 18:19:04 2022.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Introdução a linguagem Stan (*rstan*), um software para modelos bayesianos.



Universidade Federal do Oeste do Pará (UFOPA) Campus de Monte Alegre – Engenharia de Aquicultura

Obrigado !!!

Professor: Carlos Antônio Zarzar

E-mail: carloszarzar_@hotmail.com

carlos.zarzar@ufopa.edu.br

Data: 08/03/2022

AGRADECIMENTO E COLABORADORES:





