# MAT02034 - Métodos bayesianos para análise de dados

## Modelos bayesianos hierárquicos - exemplo: dados de mortalidade de câncer

Rodrigo Citton P. dos Reis
citton.padilha@ufrgs.br

Universidade Federal do Rio Grande do Sul
Instituto de Matemática e Estatística
Departamento de Estatística

Porto Alegre, 2022

# Introdução

# Introdução

Nesta aula, através de um exemplo, vamos discutir os seguintes temas:

- Reparametrização;
- Gibbs com passos de M-H;
- O algoritmo M-H passeio aleatório;
- Parâmetros de *tuning*;
- Computação no `R`;
- Representação gráfica de modelos bayesianos hierárquicos (DAGs).

# Exemplo: dados de mortalidade de câncer

# Dados de mortalidade de câncer

- Para ilustrar os métodos descritos nas aulas anteriores **(MCMC e MBH)**, considere os dados de mortalidade por câncer analisados por Tsutakawa, Shoop e Marienfeld (1985)[1].
- Estamos interessados em **estimar simultaneamente as taxas de óbito** por câncer de estômago para homens "em risco" na faixa etária de 45 a 64 anos para as 84 maiores cidades do Missouri.

[1] Tsutakawa RK, Shoop GL, Marienfeld CJ. Empirical Bayes estimation of cancer mortality rates. *Stat Med.* 1985 Apr-Jun;4(2):201-12. doi: 10.1002/sim.4780040210.

# Dados de mortalidade de câncer

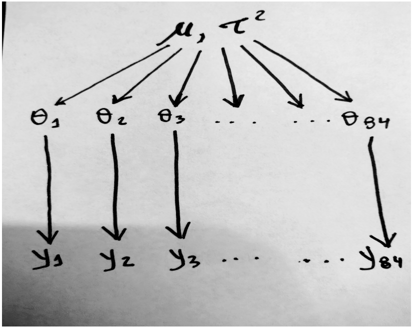
- Para a $i$-ésima cidade ($i = 1, \ldots, 84$), observa-se o número $N_i$ em risco e o número de óbitos por câncer $y_i$.
- Suponha que os $\{y_i\}$ sejam independentes com distribuições binomiais e $p_i = \Pr(Y_i = y_i)$ sejam as respectivas probabilidades de óbito.
- Se as taxas de mortalidade por câncer não variam significativamente entre as cidades (pequenas ou grandes) e entre os homens nessa faixa etária, então pode ser razoável acreditar *a priori* que as taxas são **intercambiáveis**.

# Dados de mortalidade de câncer

- Essa crença *a priori* pode ser modelada deixando os **logitos (*logit*)** $\theta_i = \log(p_i/(1-p_i))$ serem independentes $N(\mu, \tau^2)$, e então atribuindo $\mu$ e $\tau^2$ distribuições independentes $N(\mu_0, \sigma^2_\mu)$ e $GaI(a, b)$.
  - $\tau^2$ é um parâmetro de variância;
  - $GaI(a, b)$ represeeta a distribuição **gama invertida**; se $X \sim GaI(a, b)$, então $f(x|a, b) = \frac{b^a}{\Gamma(a)}(1/x)^{a+1}e^{-b/x}$.
  - $(\mu_0, \tau^2, a, b)$ são hiperparâmetros a serem especificados.

# DAGs

- Podemos exibir esquematicamente o modelo hierárquico (assim, como outras relações de dependência) por meio de um **grafo acíclico dirigido (*directed acyclic graphs*, DAGs)**.



# Dist. a posteriori

# Dist. a posteriori

- A distribuição *a posteriori* deste modelo, $g(\boldsymbol{\theta}, \mu, \tau^2, a, b|\mathbf{y})$, é proporcional à

$$
\begin{aligned}
& p(\mathbf{y}|\boldsymbol{\theta}, \mu, \tau^2) \times g_1(\boldsymbol{\theta}|\mu, \tau^2) \times g_2(\mu, \tau^2) \\
= & \left\{\prod_{i=1}^{n} p(y_i|h(\theta_i))\right\} \times \left\{\prod_{i=1}^{n} g_1(\theta_i|\mu, \tau^2)\right\} \times g_2(\mu, \tau^2) \\
= & \left\{\prod_{i=1}^{n} p(y_i|h(\theta_i)) \times g_1(\theta_i|\mu, \tau^2)\right\} \times g_2(\mu, \tau^2).
\end{aligned}
$$

- $h(\theta_i) = \frac{e^{\theta_i}}{1+e^{\theta_i}} = p_i$ é a inversa da função *logit*; ou seja, o **expito (*expit*)**.

# Dist. a posteriori

- Devido à estrutura de **independência condicional** deste modelo, é simples construir um algoritmo de simulação MCMC para gerar realizações a partir da distribuição *a posteriori* conjunta de $(\boldsymbol{\theta}, \mu, \tau^2)$.
- Condicional nos parâmetros $\mu$ e $\tau^2$, $\theta_1, \ldots, \theta_n$ têm distribuições *a posteriori* independentes com $\theta_i$ distribuída de acordo com a densidade proporcional a

$$p(y_i|h(\theta_i)) \times g_1(\theta_i|\mu, \tau^2) = Bin(y_i|N_i, \text{expit}(\theta_i)) \times \phi(\theta_i|\mu, \tau^2),$$

em que $\phi(\cdot|\mu, \tau^2)$ é a função densidade normal.

# Dist. a posteriori

- Então, combinando os estágios 2 e 3 do modelo, vê-se que os hiperparâmetros $\mu$ e $\tau^2$ possuem as seguintes distribuições condicionais:

$$\mu|\boldsymbol{\theta}, \tau^2, \mathbf{y} \sim N\left(\frac{\sigma_\mu^2}{\sigma_\mu^2 + \tau^2}\overline{\theta} + \frac{\tau^2}{\sigma_\mu^2 + \tau^2}\mu_0,\ \frac{\tau^2\sigma_\mu^2}{\sigma_\mu^2 + \tau^2}\right)$$

e

$$\tau^2|\boldsymbol{\theta}, \mu, \mathbf{y} \sim GaI\left(a + \frac{n}{2},\ b + \frac{1}{2}\sum_{i=1}^{n}[\theta_i - \mu]^2\right),$$

em que $\overline{\theta} = \frac{1}{n}\sum_{i=1}^{n}\theta_i$.

# Dist. a posteriori

# Dist. a posteriori

# Esquema MCMC

# Esquema MCMC

- Dado um valor inicial nos hiperparâmetros $\mu$ e $\tau^2$, use $n$ passos de Metropolis-Hastings independentes para simular valores das taxas transformadas $\theta_i$.
  - Podemos utilizar uma cadeia de **passeio aleatório** com a densidade de incremento $N(0, c_T)$, em que o desvio padrão $c$ é parâmetro de ***tuning***.
- Em seguida, dados os valores simulados de $\theta_i$, simule $\mu$ e $\tau^2$ das distribuições *a posteriori* gama invertida e normal, em que se está condicionando aos valores simulados mais recentemente.
- Este ciclo simula uma cadeia de Markov que converge para a distribuição *a posteriori* conjunta de $\boldsymbol{\theta}$, $\mu$ e $\tau^2$.
  - Mais uma vez estamos utilizando o **amostrador de Gibbs com passos de M-H** (aqui o **M-H passeio aleatório**).
- Após descartar um conjunto de iterações de burn-in, a amostra simulada restante é tomada como uma amostra da distribuição *a posteriori* conjunta.

# (M-H passeio aleatório)

- Seja a distribuição proposta, $q(y, x) = q(y|x)$ tal que

$$y = x + \varepsilon,$$

em que $\varepsilon$ $f$ e $f$ é uma densidade de probabilidade simétrica em 0.

- Dada esta definição, temos que

$$q(y|x) = f(\varepsilon)$$

e

$$q(x|y) = f(-\varepsilon) = f(\varepsilon).$$

# (M-H passeio aleatório)

- Porque a proposta $q(y|x)$ é simétrica em $x$ e $y$, a probabilidade de aceitação do Metropolis-Hastings $\rho(x, y)$ simplifica

$$\begin{aligned}\rho(x,y) &= \min\left\{\frac{g(y)q(x|y)}{g(x)q(y|x)}, 1\right\} \\ &= \min\left\{\frac{g(y)}{g(x)}, 1\right\}.\end{aligned}$$

# (M-H passeio aleatório)

Dado o estado atual $x$, o algoritmo de Metropolis-Hastings procede da seguinte forma:

1. Simular $\varepsilon \sim f$ e faça $y = x + \varepsilon$.
2. Calcule a probabilidade de aceitação $\rho(x, y) = \min\left\{\frac{g(y)}{g(x)}, 1\right\}$.
3. Aceite $y$ com probabilidade $\rho(x, y)$, caso contrário, continue em $x$.

Deve-se notar que esta forma do algoritmo Metropolis-Hastings era a forma original do **algoritmo Metropolis**.

# Implementação em R

# Implementação em R

- Considere o uso do algoritmo MCMC básico descrito na seção anterior com os valores dos hiperparâmetros definidos em $\mu_0 = 0$, $\sigma^2_\mu = 1.000$, $a = 5$ e $b = 1$.
    - Esses valores refletem crenças *a priori* relativamente **vagas** sobre a localização dos parâmetros de segundo estágio.
- A seguir, carregamos os dados do arquivo `Tsutakawa1985.txt`.

# Implementação em R

```
library(readr)
library(dplyr)

taxasMiss <- read_delim(
  file = here::here("dados",
                    "Tsutakawa1985.txt"),
  delim = "\t", escape_double = FALSE,
  trim_ws = TRUE)

taxasMiss
```

```
## # A tibble: 84 x 3
##    Cidade     N     y
##     <dbl> <dbl> <dbl>
##  1      1 98066    99
##  2      2 53637    54
##  3      3 46394    80
##  4      4 12890    17
##  5      5 10975    11
##  6      6  7436    13
##  7      7  3814     3
##  8      8  3461     2
```

# Implementação em R

```
##  9      9  3349     1
## 10     10  3215     1
## # ... with 74 more rows
```

# Implementação em R

```
library(invgamma)

# Configuração

mu_0 <- 0
sigma2_mu <- 1000
a <- 5
b <- 1
c_T <- 1.5

n <- length(taxasMiss$y)
M <- 10000
burnin <- 10000
k <- 10

cadeia_post <- matrix(0, nrow = M, ncol = n + 2)
colnames(cadeia_post) <- c(paste0("theta_", 1:n), "mu", "tau2")

# Valores iniciais
theta <- log( (taxasMiss$y + 1/2)/(taxasMiss$N - taxasMiss$y + 1/2) )
mu <- 0
tau2 <- 1

cadeia_post[1,] <- c(theta, mu, tau2)
```

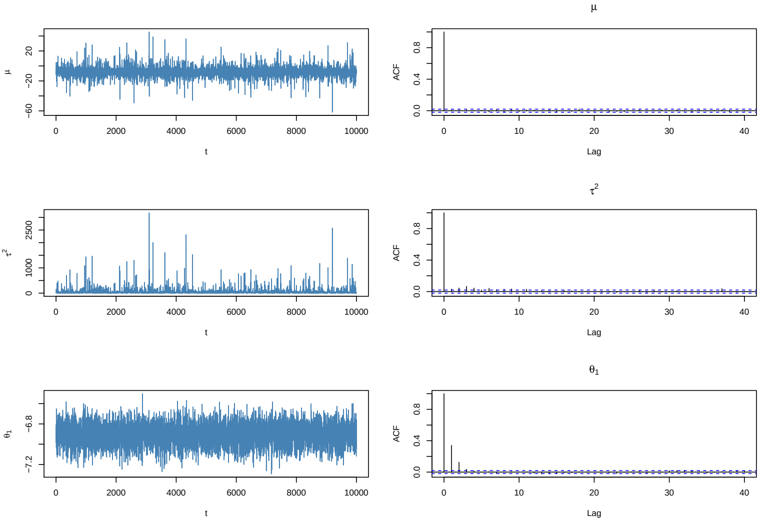
# Implementação em R

```
# Amostrador de Gibbs (burnin)

for (t in 1:burnin){

  # Passos de M-H passeio aleatório
  theta_cand <- theta +
    rnorm(n = n, mean = rep(0, n), sd = rep(c_T, n))

  razao_a <- ( dbinom(x = taxasMiss$y, size = taxasMiss$N,
    prob = (exp(theta_cand)/(1 + exp(theta_cand))) ) *
    dnorm(x = theta_cand, mean = mu, sd = sqrt(tau2)) )/
    ( dbinom(x = taxasMiss$y, size = taxasMiss$N,
    prob = (exp(theta)/(1 + exp(theta))) ) *
    dnorm(x = theta, mean = mu, sd = sqrt(tau2)) )

  rho <- apply(X = data.frame(razao_a),
               MARGIN = 1,
               FUN = function(x){min(x, 1)})
  u <- runif(n = n, min = 0, max = 1)
  theta <- ifelse(u <= rho, theta_cand, theta)

  mu <- rnorm(n = 1,
              mean = (sigma2_mu/(sigma2_mu + tau2)) * mean(theta) +
                (sigma2_mu/(sigma2_mu + tau2)) * mu_0,
              sd = sqrt( (sigma2_mu*tau2)/(sigma2_mu + tau2) ) )

  tau2 <- rinvgamma(n = 1,
                    shape = a + n/2,
                    rate = b + (1/2) * sum( (theta - mu)^2 ))
}
```

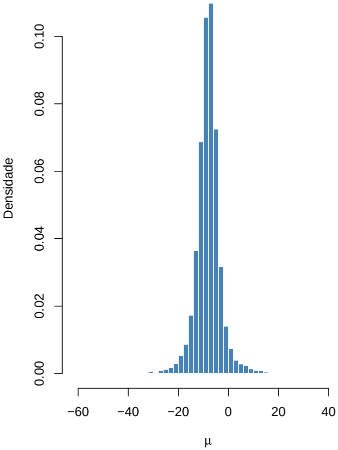
# Implementação em R

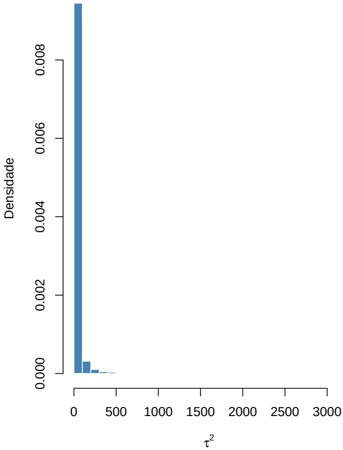
```
for (t in 1:(k*M)){
  # Passos de M-H passeio aleatório
  theta_cand <- theta +
    rnorm(n = n, mean = rep(0, n), sd = rep(c_T, n))

  razao_a <- ( dbinom(x = taxasMiss$y, size = taxasMiss$N,
    prob = (exp(theta_cand)/(1 + exp(theta_cand))) ) *
    dnorm(x = theta_cand, mean = mu, sd = sqrt(tau2)) )/
    ( dbinom(x = taxasMiss$y, size = taxasMiss$N,
    prob = (exp(theta)/(1 + exp(theta))) ) *
    dnorm(x = theta, mean = mu, sd = sqrt(tau2)) )

  rho <- apply(X = data.frame(razao_a),
               MARGIN = 1,
               FUN = function(x){min(x, 1)})
  u <- runif(n = n, min = 0, max = 1)
  theta <- ifelse(u <= rho, theta_cand, theta)

  mu <- rnorm(n = 1,
              mean = (sigma2_mu/(sigma2_mu + tau2)) * mean(theta) +
                (sigma2_mu/(sigma2_mu + tau2)) * mu_0,
              sd = sqrt( (sigma2_mu*tau2)/(sigma2_mu + tau2) ) )

  tau2 <- rinvgamma(n = 1,
                    shape = a + n/2,
                    rate = b + (1/2) * sum( (theta - mu)^2 ))

  if (t %% k == 0) {cadeia_post[(t/k),] <- c(theta, mu, tau2)}
}
```

# Implementação em R

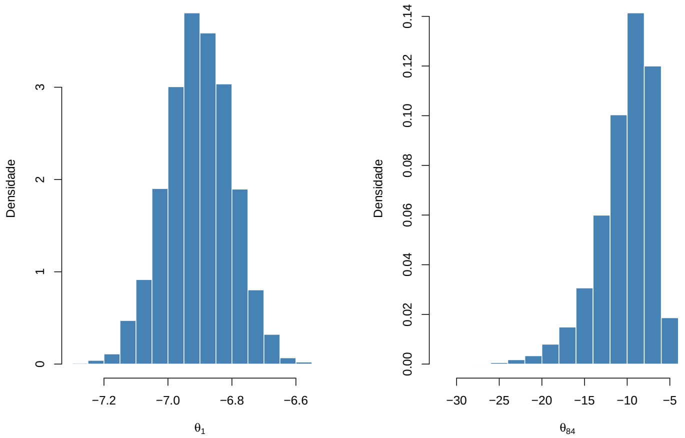


# Implementação em R



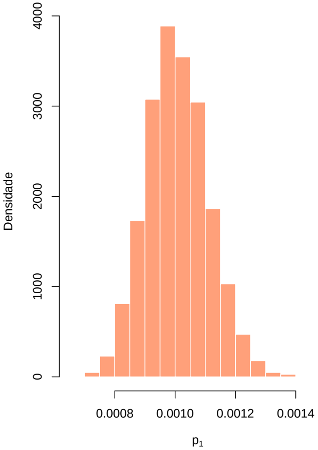



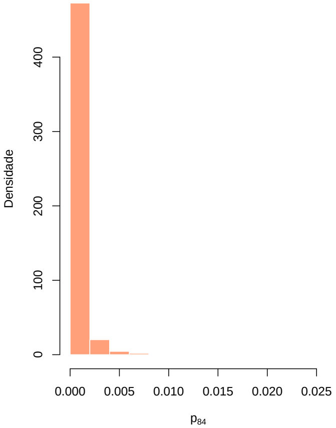
# Implementação em R



# Implementação em R

- E os $p_i$?



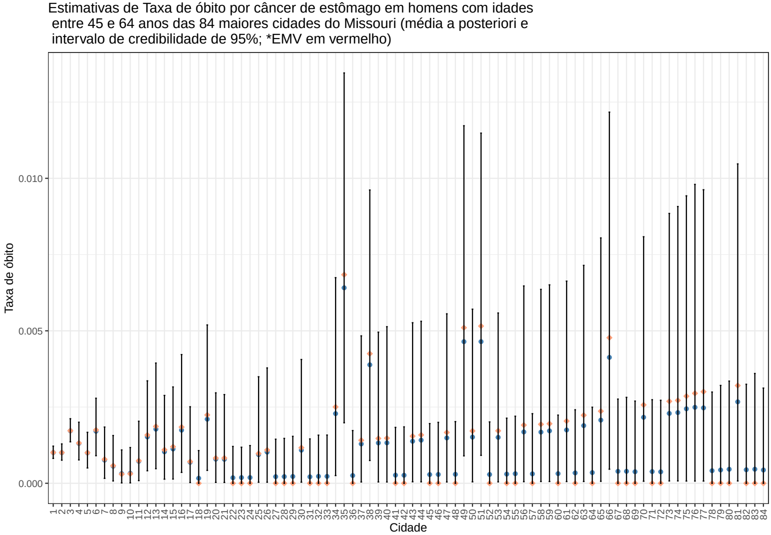
# Implementação em R

```
# Resumo da distribuição
# a posteriori das taxas de scram

p_post <- data.frame(
  cidade = factor(taxasMiss$Cidade),
  y = taxasMiss$y,
  N = taxasMiss$N,
  EMV = taxasMiss$y/taxasMiss$N,
  media = colMeans( exp(cadeia_post[,1:n])/(1 + exp(cadeia_post[,1:n])) ) )

p_post$li95 <- apply(X = exp(cadeia_post[,1:n])/(1 + exp(cadeia_post[,1:n])),
                          MARGIN = 2,
                          FUN = quantile, probs = 0.025)

p_post$ls95 <- apply(X = exp(cadeia_post[,1:n])/(1 + exp(cadeia_post[,1:n])),
                          MARGIN = 2,
                          FUN = quantile, probs = 0.975)

head(p_post)
```

# Implementação em R

```
##         cidade  y     N         EMV        media         li95        ls95
## theta_1      1 99 98066 0.001009524 0.0010078268 0.0008151985 0.001218555
## theta_2      2 54 53637 0.001006768 0.0010035848 0.0007577165 0.001291516
## theta_3      3 80 46394 0.001724361 0.0017197524 0.0013602576 0.002117783
## theta_4      4 17 12890 0.001318852 0.0013076135 0.0007665179 0.002002235
## theta_5      5 11 10975 0.001002278 0.0009972582 0.0005024144 0.001671299
## theta_6      6 13  7436 0.001748252 0.0017082838 0.0009047188 0.002789167
```

# Implementação em R



# Implementação em R

- Note que para as cidades menores, estimativas mais precisas são obtidas quando estas tomam emprestado informações das demais cidades.
  - Este é chamado de **"borrowing strength"** nos modelos bayesianos hierárquicos.

# Para casa

- Rodar os códigos dos exemplos de aula.
    - Trazer as dúvidas para o Fórum Geral do Moodle e para a próxima aula.

# Próxima aula

- OpenBugs e Jags.

# Por hoje é só!

## Bons estudos!