

# Estatística Bayesiana

Jose Storopoli

josees@uni9.pro.br Universidade Nove de Julho - UNINOVE

# Sumário

1. Estatística Bayesiana
2. Distribuições Probabilísticas
3. `rstanarm` e `brms`
4. *Priors*
5. Verificações Preditivas
6. Regressão Linear
7. Regressão Logística
8. Regressão de Poisson
9. Regressão Robusta
10. Modelos Multiníveis
11. *Markov Chain Monte Carlo - MCMC*
12. Comparação de Modelos
13. Referências

# Sumário para Estatística Bayesiana

## 1.1 Leituras Recomendadas

## 1.2 O que é Estatística Bayesiana

### 1.2.1 O que muda da Estatística Frequentista?

## 1.3 Ferramentas

### 1.3.1 Stan

### 1.3.2 R

### 1.3.3 Python

### 1.3.4 Julia

## 1.4 Probabilidade

### 1.4.1 O que é Probabilidade?

### 1.4.2 Teorema de Bayes

## 1.5 Estatística Frequentista versus Bayesiana

### 1.5.1 O que são $p$ -valores e Intervalos de Confiança

## 1.6 Estatística Bayesiana

### 1.6.1 Vantagens da Estatística Bayesiana

# Estatística Bayesiana - Leituras Recomendadas

- Gelman et al. (2013b) - Capítulo 1: Probability and inference
- McElreath (2020) - Capítulo 1: The Golem of Prague
- Gelman, Hill e Vehtari (2020) - Capítulo 3: Some basic methods in mathematics and probability
- Khan e Rue (2021)
- Storopoli (2021) - O que é Estatística Bayesiana?
- **Probabilidade:**
  - Um ótimo livro-texto - Bertsekas e Tsitsiklis (2008)
  - Um ótimo livro-texto (pule a parte de estatística frequentista) - Dekking et al. (2010)
  - Do ponto de vista Bayesiano e com abordagem filosófica - Jaynes (2003)
  - Do ponto de vista Bayesiano e com abordagem simples e lúdica - Kurt (2019)
  - Abordagem filosófica e uma exposição não focada no rigor matemático - Diaconis e Skyrms (2019)

# O que é Estatística Bayesiana

A estatística Bayesiana<sup>1</sup> é uma abordagem de análise de dados baseada no teorema de Bayes, onde o conhecimento disponível sobre os parâmetros em um modelo estatístico é atualizado com as informações dos dados observados (Gelman et al., 2013b). O conhecimento prévio é expresso como uma distribuição *a priori*<sup>2</sup> e combinado com os dados observados na forma de uma função de verossimilhança<sup>3</sup> para determinar a distribuição posterior<sup>4</sup>. A posterior também pode ser usada para fazer previsões sobre eventos futuros.

---

<sup>1</sup>maiúsculo, pois se refere ao teorema de Bayes que é um sobrenome

<sup>2</sup>do inglês *prior distribution*

<sup>3</sup>do inglês *likelihood function*

<sup>4</sup>do inglês *posterior distribution*

# O que muda da Estatística Frequentista?

- **Flexibilidade** - peças probabilísticas para construir um modelo<sup>5</sup>:
  - Conjecturas probabilísticas sobre os parâmetros:
    - *Priori*
    - Verossimilhança
- Melhor tratamento da **incerteza**:
  - Coerência
  - Propagação
  - Não se usa "*se amostrássemos infinitamente de uma população que não existe...*"
- Sem **p-valores**:
  - Todas as intuições estatísticas fazem **sentido**
  - 95% de certeza que o valor do parâmetro  $\theta$  está entre  $x$  e  $y$
  - Quase **impossível** fazer *p-hacking*.

---

<sup>5</sup>como se fosse LEGO

# Um pouco mais de Formalidade

- Estatística Bayesiana usa declarações probabilísticas:
  - um ou mais parâmetros  $\theta$
  - dados não-observados  $\tilde{y}$
- Essas declarações são condicionadas nos valores observados de  $y$ :
  - $P(\theta | y)$
  - $P(\tilde{y} | y)$
- Nós também, de maneira implícita, condicionados nos valores observados de quaisquer co-variáveis  $x$

# Principal Mudança

## Definição (Estatística Bayesiana)

*O uso do Teorema de Bayes<sup>6</sup> como o procedimento de **estimativa dos parâmetros de interesse  $\theta$  ou dados não-observados  $\tilde{y}$ .** (Gelman et al., 2013b)*

---

<sup>6</sup>mais sobre ele já já...

# Ferramentas para Estatística Bayesiana

- Stan
- PyMC
- JAGS
- BUGS

# Stan<sup>7</sup>

- Plataforma para modelagem e computação estatística de alto desempenho
- Suporte financeiro da NUMFocus:
  - AWS Amazon
  - Bloomberg
  - Microsoft
  - IBM
  - RStudio
  - Facebook
  - NVIDIA
  - Netflix
- Linguagem própria, similar à C++
- Amostrador *Markov Chain Monte Carlo* (MCMC) em paralelo



---

<sup>7</sup>Carpenter et al. (2017)

Stan na Série Billions<sup>8</sup> (Temporada 3 Episódio 9)

# Stan na Série Billions

---

<sup>8</sup>Se não conseguir assistir clique [aqui](#) para ver o vídeo no seu navegador

- R é uma linguagem criada **por estatísticos para estatísticos**
- Possui um vasto **ecossistema de bibliotecas** e é amplamente usado na **ciência** e em especial nas **ciências aplicadas**
- Quase toda **tese maluca** ou **algoritmo inovador** de Estatística/Probabilidade está no CRAN (Repositório de Pacotes R)
- Como linguagem de programação é **horrível**<sup>9</sup>: Recomendo Julia

---

<sup>9</sup>consegue ser um pouco menos pior que Python

# Python e PyMC

- Python consegue ser um pouco melhor que R
- Mas tem a "tara" dos anos 90 de tudo ser **Orientado à Objetos**
- PyMC (Salvatier et al., 2016):
  - Uma Biblioteca de Estatística Bayesiana com o seu próprio amostrador *Markov Chain Monte Carlo* (MCMC)
  - Também com Suporte financeiro da NUMFocus
  - Amarram o cavalo num barco que afundou há algum tempo: Theano
  - Theano **morreu** mas os desenvolvedores do PyMC fizeram um *fork* no projeto e estão usando-o como *backend*

# Julia e Turing

- Julia (Bezanson et al., 2017) é uma linguagem relativamente nova, lançada pela primeira vez em 2012, que visa ser de **alto nível e rápida**
- Linguagem de tipagem **dinâmica rápida** que compila *just-in-time* (JIT) em código nativo usando LLVM.
- "Roda como C, mas lê como Python" (Perkel, 2019), o que significa que é extremamente **rápida**, fácil **prototipagem e ler/escrever** código.
- **Multi-paradigma**, combinando recursos de programação **imperativa, funcional e orientada a objetos**.

# Turing

- Turing é uma **Linguagem de Programação Probabilística**<sup>10</sup> escrita totalmente em Julia
- Usa **pacotes** de Julia para:
  - **Diferenciação Automática**<sup>11</sup>
  - **Distribuições Probabilísticas**
  - **Solucionadores de Equações Ordinais**<sup>12</sup>
  - **Redes Neurais** (sendo responsável pela parte "Bayesiana" da Rede Neural Bayesiana)

---

<sup>10</sup>em inglês *probabilistic programming language* (PPL)

<sup>11</sup>*autodiff*

<sup>12</sup>*Ordinary Differential Equation Solvers* (ODE)

# PROBABILIDADE NÃO EXISTE!<sup>13</sup>

- Sim, a probabilidade não existe.
- Ou melhor, probabilidade como uma quantidade física, chance objetiva, **NÃO existe**
- se dispensarmos a questão da chance objetiva *nada se perde*
- A matemática do raciocínio indutivo permanece **exatamente a mesma**

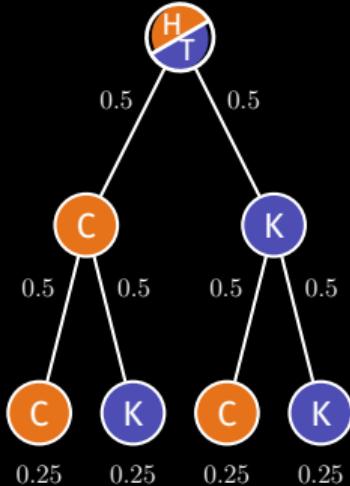


---

<sup>13</sup>de Finetti (1974)

# PROBABILIDADE NÃO EXISTE!<sup>14</sup>

- Considere jogar uma moeda de enviesada
- As tentativas são consideradas independentes e, como resultado, exibem outra propriedade importante: **a ordem não importa**
- A frequência é considerada uma **estatística suficiente**
- Dizer que a ordem não importa ou dizer que a única coisa que importa é a frequência são duas maneiras de dizer exatamente a mesma coisa
- Dizemos que essa probabilidade é **invariante sob permutações**



<sup>14</sup>de Finetti (1974)

# Interpretações da Probabilidade

- **Objetiva** - frequência no longo prazo de um evento específico

- $P(\text{chuva}) = \frac{\text{dias que choveram}}{\text{dias totais}}$

- $P(\text{chance de eu ser presidente} = 0)$  (Nunca ocorreu)

- **Subjetiva** - nível de crença em um evento

- $P(\text{chuva}) = \text{crença que choverá}$

- $P(\text{chance de eu ser presidente} = 10^{-10})$  (Muito improvável)

# O que é Probabilidade?

## Definição (Probabilidade)

*Sobre notação, definimos que  $A$  é um evento e  $P(A)$  a probabilidade do evento, logo:*

$$\{P(A) \in \mathbb{R} : 0 \leq P(A) \leq 1\}.$$

*Isto quer dizer o "probabilidade do evento  $A$  ocorrer é o conjunto de todos os números reais entre 0 e 1; incluindo 0 e 1"*

# Axiomas da Probabilidade<sup>15</sup>

- **Não-negatividade:** Para todo  $A$ ,  $P(A) \geq 0$ . Toda probabilidade é positiva (maior ou igual a zero), independente do evento
- **Aditividade:** Para dois *mutuamente exclusivos*  $A$  e  $B$  (não podem ocorrer ao mesmo tempo):  
 $P(A) = 1 - P(B)$  e  $P(B) = 1 - P(A)$
- **Normalização:** A probabilidade de todos os eventos possíveis  $A_1, A_2, \dots$  devem somar 1:  $\sum_{n \in \mathbb{N}} A_n = 1$



---

<sup>15</sup>Kolmogorov (1933)

# Espaços Amostrais

- Discretos

$$\Theta = \{1, 2, \dots, \}$$

- Contínuos

$$\Theta \in (-\infty, \infty)$$

# Espaços Amostrais Discretos

## 8 Planetas do Nossa Sistema Solar

- Mercúrio - ♀
- Vênus - ♀
- Terra - ♂
- Marte ♂
- Júpiter - ♀
- Saturno ♂
- Urano - ♂
- Netuno ♂

# Espaços Amostrais Discretos<sup>16</sup>

O planeta possui campo magnético



O planeta possui luas



O planeta possui campo magnético e luas



O planeta possui campo magnético ou luas



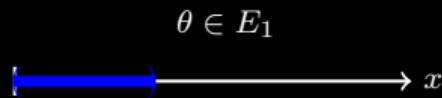
O planeta não possui um campo magnético



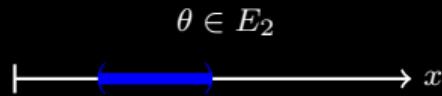
<sup>16</sup>figuras adaptadas de Michael Betancourt (CC-BY-SA-4.0)

# Espaços Amostrais Contínuos<sup>17</sup>

A distância é menos que cinco centímetros



A distância é entre três e sete centímetros



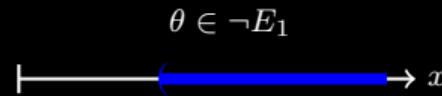
A distância é menos que cinco centímetros  
e entre três e sete centímetros



A distância é menos que cinco centímetros  
ou entre três e sete centímetros



A distância não é menos que cinco centímetros



<sup>17</sup>figuras adaptadas de Michael Betancourt (CC-BY-SA-4.0)

# Parâmetros Discretos versus Contínuos

Tudo o que foi exposto até agora partiu do pressuposto que os parâmetros são discretos. Isto foi feito com o intuito de prover uma melhor intuição do que é probabilidade. Nem sempre trabalhamos com parâmetros discretos. Os parâmetros podem ser contínuos, como por exemplo: idade, altura, peso etc. Mas não se desespere, todas as regras e axiomas da probabilidade são válidos também para parâmetros contínuos. A única coisa que temos que fazer é trocar todas as somas  $\sum$  por integrais  $\int$ . Por exemplo o terceiro axioma de **Normalização** para variáveis aleatórias contínuas se torna:

$$\int_{x \in X} p(x)dx = 1.$$

# Probabilidade Condicional

## Definição (Probabilidade Condicional)

*Probabilidade de um evento ocorrer caso outro tenha ocorrido ou não.*

*A notação que usamos é  $P(A | B)$ , que  
lê-se como "a probabilidade de observamos A dado que já observamos B".*

$$P(A | B) = \frac{\text{número de elementos em } A \text{ e } B}{\text{número de elementos em } B}$$

$$P(A | B) = \frac{P(A \cap B)}{(B)}$$

# Exemplo de Probabilidade Condicional

## Exemplo (Poker Texas Hold'em)

- **Espaço Amostral:** 52 cartas no baralho, 13 tipos de cartas e 4 tipos de naipes.
- $P(A)$ : Chance de receber um Ás ( $\frac{4}{52} = \frac{1}{13}$ )
- $P(K)$ : Chance de receber um Rei (K) ( $\frac{4}{52} = \frac{1}{13}$ )
- $P(A | K)$ : Chance de receber um Ás, dado que você recebeu um Rei (K) ( $\frac{4}{51} \approx 0.078$ )
- $P(K | A)$ : Chance de receber um Rei (K), dado que você recebeu um Ás ( $\frac{4}{51} \approx 0.078$ )

# Cuidado! Nem sempre $P(A | B) = P(B | A)$

No exemplo anterior temos a simetria  $P(A | K) = P(K | A)$ , **mas nem sempre isso é verdade**<sup>18</sup>

## Exemplo (O Papa é católico)

- $P(\text{papa})$ : Chance alguém aleatório ser papa, algo bem pequeno, 1 em 8 bilhões ( $\frac{1}{8 \cdot 10^9}$ )
- $P(\text{católico})$ : Chance alguém aleatório ser católico, 1.34 de 8 bilhões ( $\frac{1.34}{8} \approx 0.17$ )
- $P(\text{católico} | \text{papa})$ : Chance do Papa ser católico ( $\frac{999}{1000} = 0.999$ )
- $P(\text{papa} | \text{católico})$ : Chance de alguém católico ser o papa ( $\frac{1}{1.34 \cdot 10^9} \cdot 0.999 \approx 7.46 \cdot 10^{-10}$ )
- **Logo:**  $P(\text{católico} | \text{papa}) \neq P(\text{papa} | \text{católico})$

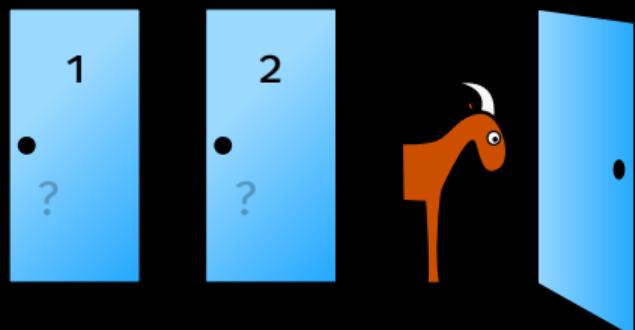
---

<sup>18</sup>Mais especificamente, se as taxas basais  $P(A)$  e  $P(B)$  não são iguais, a simetria é quebrada  
 $P(A | B) \neq P(B | A)$ !

# Um clássico da Probabilidade

## Exemplo (Monty Hall)

- Um apresentador de TV lhe apresenta 3 portas
- Uma delas tem um prêmio: um carro! As outras tem um bode
- Você deve escolher uma porta (que não é aberta)
- Nesse momento Monty abre uma das outras duas portas que você não escolheu, revelando que o carro não se encontra nessa porta e revelando um dos bodes
- Monty então lhe pergunta "Você quer manter sua escolha de porta ou trocar?"



# Solução do Problema de Monty Hall

Ideia (Probabilidade de ganhar o carro)

$$P(\text{carro} \mid C_i) = \frac{1}{3}$$

$$P(\text{carro}) = \frac{1}{3} \cdot P(\text{carro} \mid C_1) + \frac{1}{3} \cdot P(\text{carro} \mid C_2) + \frac{1}{3} \cdot P(\text{carro} \mid C_3)$$

$$P(\text{carro}) = \frac{\sum_{i=1}^3 P(\text{carro} \mid C_i)}{3}$$

$$P(\text{carro}) = \frac{1}{3}$$

$C_i$  é o evento no qual o carro está atrás da porta  $i$ ,  $i = 1, 2, 3$

# Solução do Problema de Monty Hall<sup>19</sup>

## Cenário 1: Não trocar de porta

Simples:

$$\frac{1}{3}$$

## Cenário 2: Trocar de porta

Escolha qualquer porta  $i$  para ser  
 $C_i = 0$

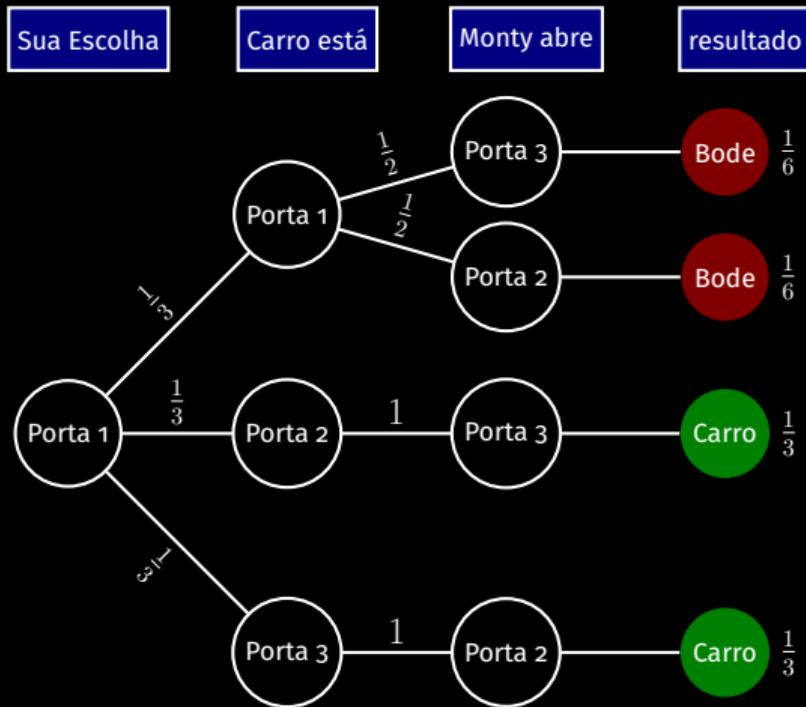
$$P(\text{carro}) = 0 \cdot P(\text{carro} \mid C_i) + \frac{1}{3} + \frac{1}{3}$$

$$P(\text{carro}) = \frac{2}{3}$$

---

<sup>19</sup>se você não acredita nesse resultado veja como simular o problema de Monty Hall nos Slides de Backup no final dessa apresentação

# Visualização do Problema de Monty Hall



# Probabilidade Conjunta

## Definição (Probabilidade Conjunta)

*Probabilidade de observados dois ou mais eventos ocorrem.*

*A notação que usamos é  $P(A, B)$ , que lê-se como "a probabilidade de observamos A e também observamos B".*

$$P(A, B) = \text{número de elementos em } A \text{ ou } B$$

$$P(A, B) = P(A \cup B)$$

# Exemplo de Probabilidade Conjunta

## Exemplo (Revisitando Poker Texas Hold'em)

- **Espaço Amostral:** 52 cartas no baralho, 13 tipos de cartas e 4 tipos de naipes.
- $P(A)$ : Chance de receber um Ás ( $\frac{4}{52} = \frac{1}{13}$ )
- $P(K)$ : Chance de receber um Rei (K) ( $\frac{4}{52} = \frac{1}{13}$ )
- $P(A | K)$ : Chance de receber um Ás, dado que você recebeu um Rei (K) ( $\frac{4}{51} \approx 0.078$ )
- $P(K | A)$ : Chance de receber um Rei (K), dado que você recebeu um Ás ( $\frac{4}{51} \approx 0.078$ )
- $P(A, K)$ : Chance de receber um Ás e um Rei (K)

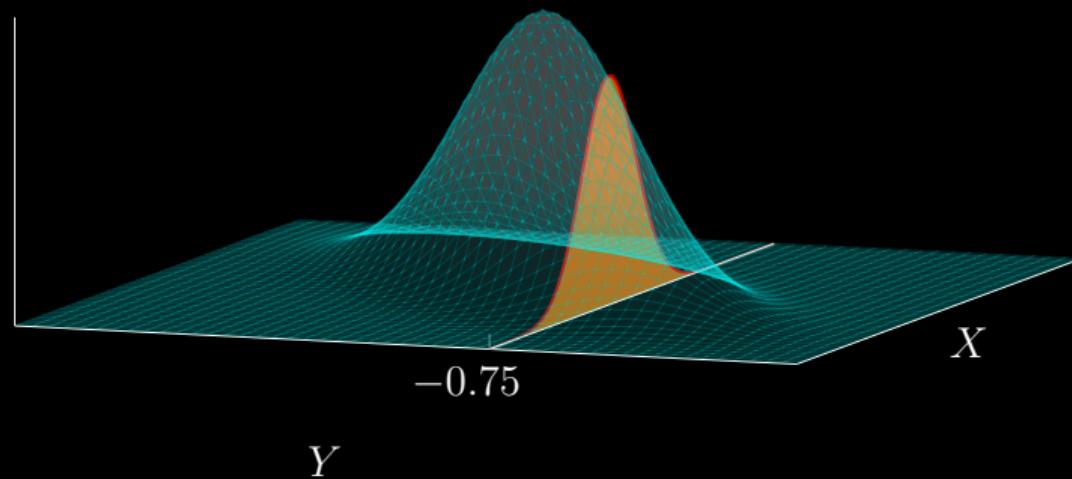
$$P(A, K) = P(K, A)$$

$$P(A) \cdot P(K | A) = P(K) \cdot P(A | K)$$

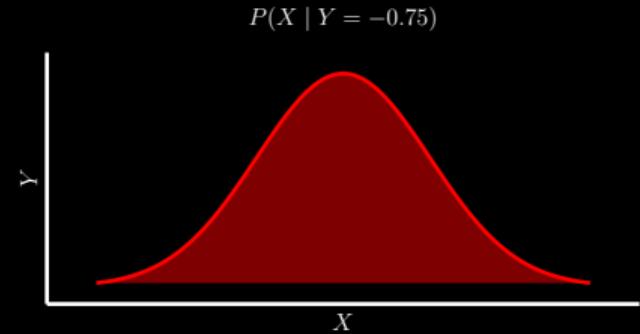
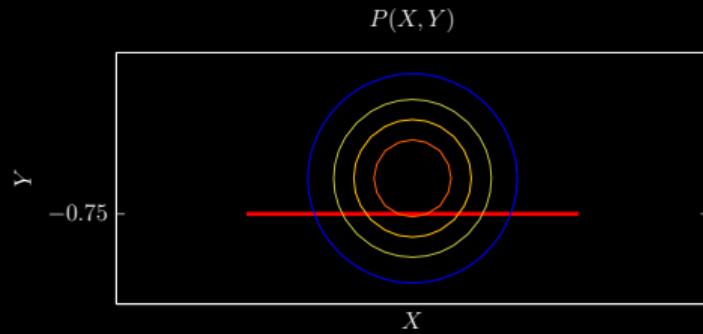
$$\begin{aligned}\frac{1}{13} \cdot \frac{4}{51} &= \frac{1}{13} \cdot \frac{4}{51} \\ &\approx 0.006\end{aligned}$$

# Visualização de Probabilidade Conjunta vs Probabilidade Condicional

$P(X, Y)$  versus  $P(X | Y = -0.75)$



# Visualização de Probabilidade Conjunta vs Probabilidade Condicional



# Quem foi Thomas Bayes?

- Thomas Bayes (1701 - 1761) foi um estatístico, filósofo e ministro presbiteriano inglês conhecido por formular um caso específico do teorema que leva seu nome
- Bayes nunca publicou o que se tornaria sua realização mais famosa; suas notas foram editadas e publicadas após sua morte pelo seu amigo Richard Price
- O nome formal do teorema é Bayes-Price-Laplace, pois Thomas Bayes foi o primeiro a descobrir, Richard Price pegou seus rascunhos, formalizou em notação matemática e apresentou para a Royal Society of London, e Pierre Laplace redescobriu o teorema sem ter tido contato prévio no final do século XVIII na França ao usar probabilidade para inferência estatística com dados do Censo na era Napoleônica



# Teorema de Bayes

## Theorema (Bayes)

*Nos diz como "inverter" a probabilidade condicional:*

$$P(A | B) = \frac{P(A) \cdot P(B | A)}{P(B)}$$

# Prova do Teorema de Bayes

Lembra que temos a seguinte identidade na probabilidade:

$$P(A, B) = P(B, A)$$

$$P(A) \cdot P(B | A) = P(B) \cdot P(A | B)$$

Pois bem, agora passe o  $P(B)$  do lado direito para o lado esquerdo dividindo:

isso vai para ←

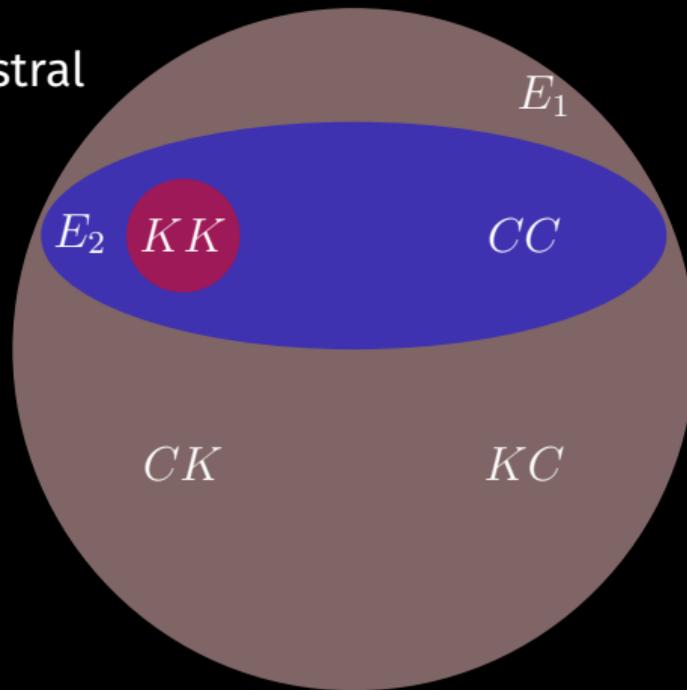
$$P(A) \cdot P(B | A) = \overbrace{P(B)}^{\leftarrow} \cdot P(A | B)$$

$$\frac{P(A) \cdot P(B | A)}{P(B)} = P(A | B)$$

$$P(A | B) = \frac{P(A) \cdot P(B | A)}{P(B)}$$

# Visualização do Teorema de Bayes

Espaço Amostral



$$E_1 = P(KK \cup CC)$$
$$E_2 = P(KK | E_1)$$

# Mais um clássico da Probabilidade<sup>20</sup>

## Exemplo (Câncer de Mama)

O quanto acurado é o teste de **câncer de mama**?

- 1% das mulheres têm **câncer de mama** (Prevalência)
- 80% das mamografias detectam o **câncer de mama** (Verdadeiro Positivo)
- 9.6% das mamografias detectam **câncer de mama** quando não há incidência (Falso Positivo)

$$P(C | +) = \frac{P(+ | C) \cdot P(C)}{P(+)}$$

$$P(C | +) = \frac{P(+ | C) \cdot P(C)}{P(+ | C) \cdot P(C) + P(+ | \neg C) \cdot P(\neg C)}$$

$$P(C | +) = \frac{0.8 \cdot 0.01}{0.8 \cdot 0.01 + 0.096 \cdot 0.99}$$

$$P(C | +) \approx 0.0776$$

---

<sup>20</sup> Origem: Yudkowsky - *An Intuitive Explanation of Bayes' Theorem*

# Porquê o teorema de Bayes é Importante?

Ideia (Podemos Inverter a Probabilidade Condisional)

$$P(\text{hipótese} | \text{dados}) = \frac{P(\text{hipótese}) \cdot P(\text{dados} | \text{hipótese})}{P(\text{dados})}$$

Mas isso não é o  $p$ -valor? **NÃO!**

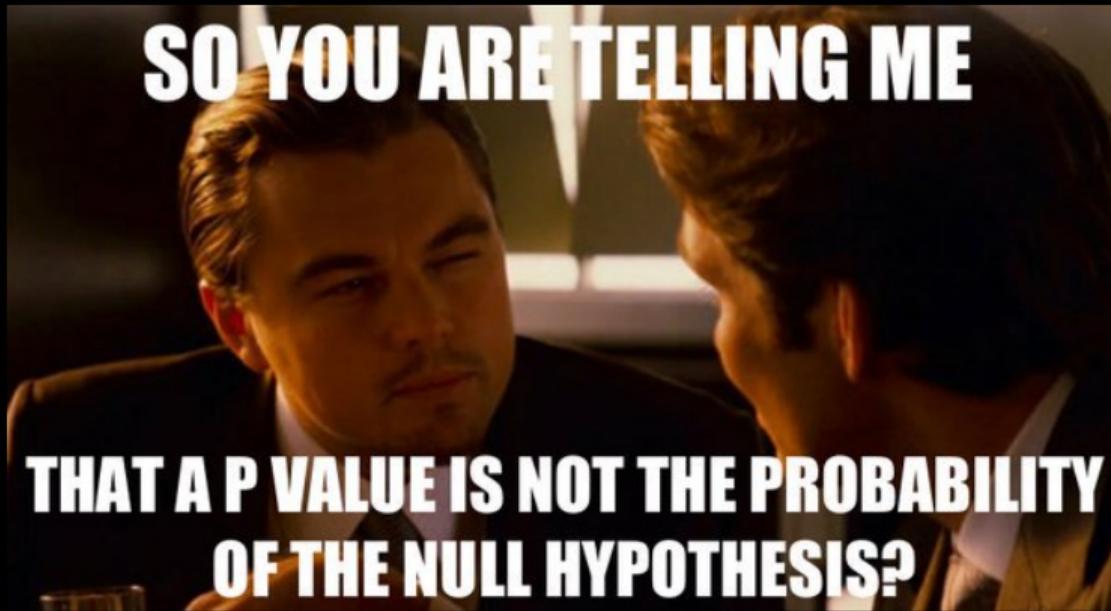
# O que é o *p*-valor?

## Definição (*p*-valor)

*p*-valor é a probabilidade de obter resultados no mínimo tão extremos quanto os que foram observados, dado que a hipótese nula  $H_0$  é verdadeira

$$P(D \mid H_0)$$

# O que **não** é o *p*-valor!



# O que **não** é o *p*-valor!

- ***p*-valor não é a probabilidade da Hipótese nula** - Famosa confusão entre  $P(D | H_0)$  e  $P(H_0 | D)$ . Para obter a  $P(H_0 | D)$  você precisa de estatística Bayesiana.
- ***p*-valor não é a probabilidade dos dados serem produzidos pelo acaso** - **Não!** Ninguém falou nada de acaso.
- ***p*-valor mensura o tamanho do efeito de um teste estatístico** - Também **não...** *p*-valor não diz nada sobre o tamanho do efeito. Apenas sobre se o quanto os dados observados divergem do esperado sob a hipótese nula. Além disso, *p*-valores podem ser "hackeados" de diversas maneiras (Head et al., 2015).

# A relação entre $p$ -valor e $H_0$

Para descobrir o  $p$ -valor, **descubra a  $H_0$  que está por trás dele**. Sua definição nunca mudará, pois ela sempre é  $P(D | H_0)$ :

- **Teste  $t$ :**  $P(D | \text{a diferença entre os grupos é zero})$
- **ANOVA:**  $P(D | \text{não há diferença entre os grupos})$
- **Régressão:**  $P(D | \text{coeficiente é nulo})$
- **Shapiro-Wilk:**  $P(D | \text{população é distribuída como uma normal})$

# O que são Intervalos de Confiança?

## Definição (Intervalos de Confiança)

*Um intervalo de confiança de X% para um parâmetro é um intervalo  $(a, b)$  gerado por um procedimento que em amostragem repetida tem uma probabilidade de X% de conter o valor verdadeiro do parâmetro, para todos os valores possíveis do parâmetro*



*Neyman (1937) (o "pai" dos intervalos de confiança)*

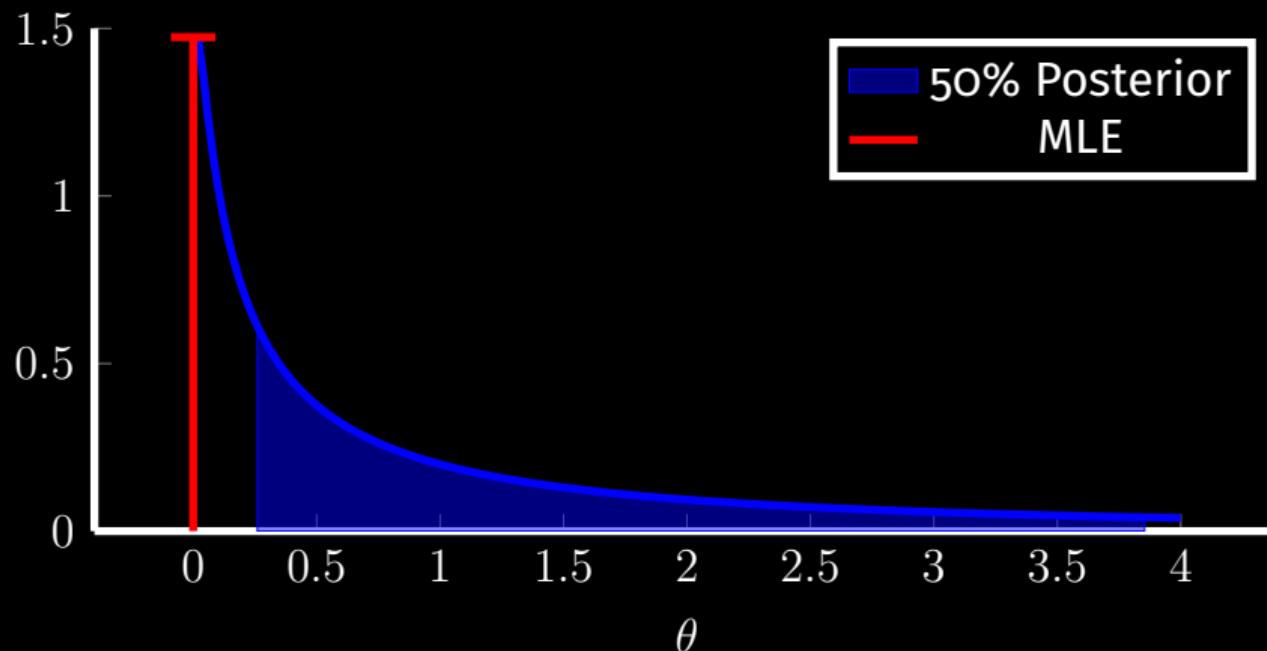
# O que são Intervalos de Confiança?

## Exemplo (Intervalo de Confiança de uma Política Pública)

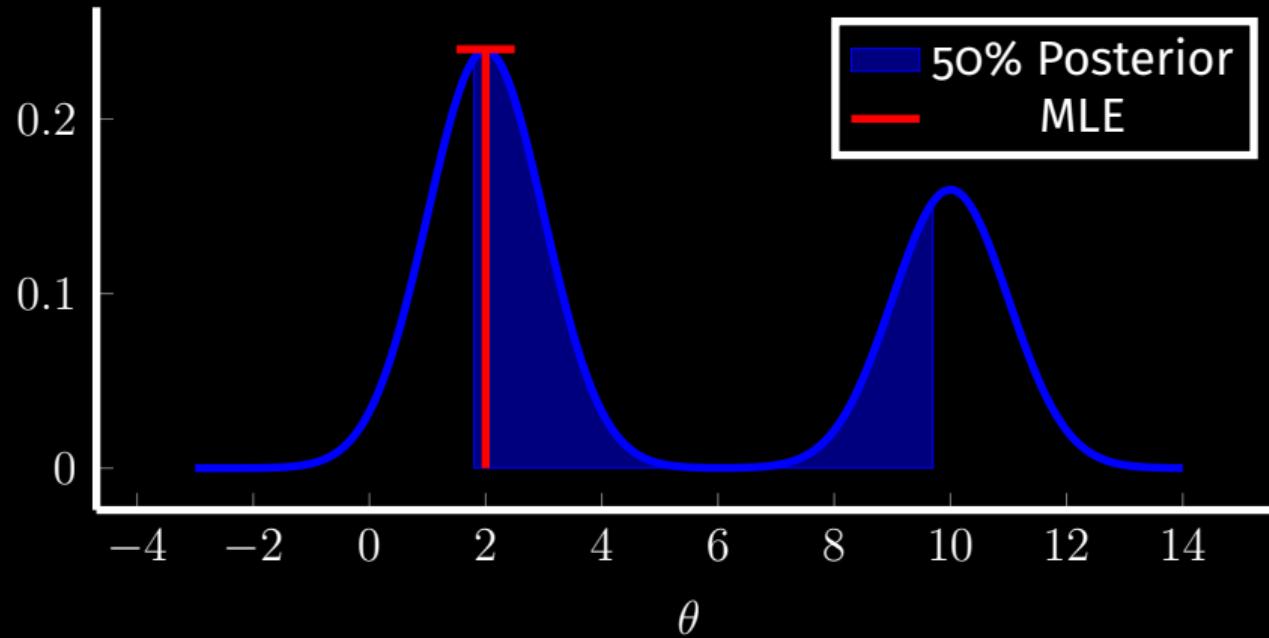
*Digamos que você executou uma análise estatística para comparar eficácia de uma política pública em dois grupos e você obteve a diferença entre a média desses grupos. Você pode expressar essa diferença como um intervalo de confiança. Geralmente escolhemos a confiança de 95%. Isso quer dizer que **95 estudos de 100**, que usem o **mesmo tamanho de amostra e população-alvo**, aplicando o **mesmo teste estatístico**, esperarão encontrar um resultado de diferenças de média entre grupos entre o intervalo de confiança.*

*Não diz nada sobre a sua **população-alvo**, mas sim sobre a sua **amostra** num processo maluco de **amostragem infinita...***

# Intervalos de Confiança versus Intervalos da Posterior



# Intervalos de Confiança versus Intervalos da Posterior



# Mas por quê eu nunca vejo estatística sem *p*-valor?

Não tem como entendermos *p*-valores se não compreendermos as suas origens e trajetória histórica. A primeira menção do termo foi feita pelo estatístico Ronald Fisher em 1925 (Fisher, 1925):

*[*p*-valor é] índice que mede a força da evidência contra a hipótese nula*



- Para quantificar a força da evidência contra a hipótese nula, Fisher defendeu " $p < 0.05$  como um nível padrão para concluir que há evidência contra a hipótese testada"
- "Não seremos frequentemente perdidos se traçarmos uma linha convencional de 0.05"

$p = 0.06$

- Como o  $p$ -valor é uma probabilidade, ele é uma quantidade contínua.
- Não há razão para diferenciarmos um  $p$  de 0.049 contra um  $p$  de 0.051.
- Robert Rosenthal, um psicólogo já dizia "Deus ama  $p$  de 0.06 tanto quanto um  $p$  de 0.05" (Rosnow & Rosenthal, 1989).

# Mas por quê eu nunca ouvi falar de Estatística Bayesiana?<sup>21</sup>

*... it will be sufficient ... to reaffirm my personal conviction ... that the theory of inverse probability is founded upon an error, and must be wholly rejected.*

Fisher (1925)



---

<sup>21</sup>*inverse probability* é como o teorema de Bayes era chamado no começo do século XX

# Dentro de todo não Bayesiano há um Bayesiano querendo sair<sup>22</sup>

- No último ano de sua vida, Fisher publicou um artigo (Fisher, 1962) examinando as possibilidades dos métodos Bayesianos, mas com as probabilidades *a priori* a serem determinadas experimentalmente.
- Inclusive alguns autores especulam (Jaynes, 2003) que se Fisher estivesse vivo hoje, ele provavelmente seria um "Bayesiano".



<sup>22</sup> Dennis Lindley "Inside every nonBayesian there is a Bayesian struggling to get out"

# Teorema de Bayes como Motor de Inferência

Agora que você já sabe o que é probabilidade e o que é o teorema de Bayes, vou propor o seguinte modelo:

$$\underbrace{P(\theta | y)}_{\text{Posterior}} = \frac{\overbrace{P(y | \theta)}^{\text{Verossimilhança}} \cdot \overbrace{P(\theta)}^{\text{Priori}}}{\underbrace{P(y)}_{\text{Constante Normalizadora}}}$$

- $\theta$  – parâmetro(s) de interesse
- $y$  – dados observados
- **Priori**: probabilidade prévia do valor do(s) parâmetro(s)
- **Verossimilhança**: probabilidade dos dados observados condicionados aos valores do(s) parâmetro(s)
- **Posterior**: probabilidade posterior do valor do(s) parâmetros após observarmos os dados  $y$
- **Constante Normalizadora**:  $P(y)$  não faz sentido intuitivo. Essa probabilidade é transformada e pode ser interpretada como algo que existe apenas para que o resultado de  $P(y | \theta)P(\theta)$  seja algo entre 0 e 1 – uma probabilidade válida.

# Teorema de Bayes como Motor de Inferência

A estatística Bayesiana nos permite **quantificar diretamente a incerteza** relacionada ao valor de um ou mais parâmetros do nosso modelo condicionado aos dados observados. Isso é a **característica principal** da estatística Bayesiana. Pois estamos estimando diretamente  $P(\theta | y)$  por meio do teorema de Bayes. A estimativa resultante é totalmente intuitiva: simplesmente quantifica a intercerteza que temos sobre o valor de um ou mais parâmetro condicionado nos dados, nos pressupostos do nosso modelo (verossimilhança) e na probabilidade prévia que temos sobre tais valores.

# Estatística Bayesiana vs Frequentista

	Estatística Bayesiana	Estatística Frequentista
<b>Dados</b>	Fixos – Não Aleatórios	Incertos – Aleatórios
<b>Parâmetros</b>	Incertos – Aleatórios	Fixos – Não Aleatórios
<b>Inferência</b>	Incerteza sobre o valor do parâmetro	Incerteza sobre um processo de amostragem de uma população infinita
<b>Probabilidade</b>	Subjetiva	Objetiva (mas com diversos pressupostos dos modelos)
<b>Incerteza</b>	Intervalo de Credibilidade – $P(\theta   y)$	Intervalo de Confiança – $P(y   \theta)$

# Vantagens da Estatística Bayesiana

- Abordagem Natural para expressar Incerteza
- Habilidade de incorporar Informações Prévias
- Maior Flexibilidade do Modelo
- Distribuição Posterior completa dos Parâmetros
- Propagação Natural da Incerteza

**Principal Desvantagem:** Velocidade lenta de estimativa de modelos<sup>23</sup>

---

<sup>23</sup>e.g. 30 segundos ao invés de 3 segundos na abordagem frequentista

# O começo do fim da Estatística Frequentista

- Saiba que você está em um momento da história no qual a Estatística está passando por grandes mudanças
- Acredito que a estatística frequentista, em especial a maneira que qualificamos evidências e hipóteses com  $p$ -valores se transformará de maneira "significante".
- Há cinco anos atrás, a *American Statistical Association* (ASA) publicou uma declaração sobre  $p$ -valores (Wasserstein & Lazar, 2016). A declaração diz exatamente o que falamos aqui: Os conceitos principais do teste de significância de hipótese nula e, em particular  $p$ -valores não conseguem prover o que os pesquisadores requerem deles. Apesar do que dizem muitos livros de estatística, materiais de ensinos e artigos publicados,  $p$ -valores abaixo de 0,05 não "provam" a realidade de nada. Nem, chegando a esse ponto, os  $p$ -valores acima de 0,05 refutam alguma coisa.
- A declaração da ASA tem mais de 3.600 citações provocando impacto relevante.

# O começo do fim da Estatística Frequentista

- Um simpósio internacional foi promovido em 2017 que originou uma edição especial de acesso aberto da *The American Statistician* dedicada à maneiras práticas de abandonarmos  $p < 0.05$  (Wasserstein et al., 2019).
- Logo na sequência vieram mais tentativas e reivindicações. Em setembro de 2017, a *Nature Human Behaviour* publicou um editorial propondo que o nível de significância do  $p$ -valor seja reduzido de 0.05 para 0.005 (Benjamin et al., 2018). Diversos autores, inclusive muitos estatísticos altamente influentes e importantes argumentaram que esse simples passo ajudaria a combater o problema da crise de replicabilidade da ciência, que muitos acreditam ser a principal consequência do uso abusivo de  $p$ -valores (Ioannidis, 2019).
- Além disso, muitos foram um passo além e sugerem que a ciência descarte de uma vez por todas  $p$ -valores («It's Time to Talk about Ditching Statistical Significance», 2019; Lakens et al., 2018). Muitos sugerem (eu inclusive) que a principal ferramenta de inferência seja a estatística Bayesiana (Amrhein et al., 2019; Goodman, 2016; van de Schoot et al., 2021)

# Sumário para Distribuições Probabilísticas

## 2.1 Leituras Recomendadas

## 2.2 Distribuições Discretas

2.2.1 Uniforme Discreta

2.2.2 Bernoulli

2.2.3 Binomial

2.2.4 Poisson

2.2.5 Binomial Negativa

## 2.3 Distribuições Contínuas

2.3.1 Uniforme Contínua

2.3.2 Normal

2.3.3 Log-Normal

2.3.4 Exponencial

2.3.5  $t$  de Student

2.3.6 Beta

# Distribuições Probabilísticas - Leituras Recomendadas

- Dekking et al. (2010)
  - Capítulo 4: Discrete random variables
  - Capítulo 5: Continuous random variables
- Betancourt (2019)
- Storopoli (2021) - Distribuições Estatísticas

# Distribuições Probabilísticas

A estatística Bayesiana usa distribuições probabilísticas como o motor de sua inferência na elaboração dos valores dos parâmetros estimados e suas incertezas.

Imagine que distribuição probabilísticas são pequenas peças de "Lego". Podemos construir o que quisermos com essas pequenas peças. Podemos fazer um castelo, uma casa, uma cidade; literalmente o que quisermos. O mesmo é válido para modelos probabilísticos em estatística Bayesiana. Podemos construir modelos dos mais simples aos mais complexo a partir de distribuições probabilísticas e suas relações entre si.

# Distribuições Probabilísticas

## Definição (Função de Distribuição de Probabilidade)

*Uma função de distribuição de probabilidade é a função matemática que fornece as probabilidades de ocorrência de diferentes resultados possíveis para um experimento. É uma descrição matemática de um fenômeno aleatório em termos de seu espaço amostral e as probabilidades de eventos (subconjuntos do espaço amostral).*

$$P(X) : X \rightarrow \mathbb{R} \in [0, 1]$$

*Para variáveis discretas chamamos de "massa" e para variáveis contínuas chamamos de "densidade".*

# Notação Matemática

Geralmente usamos a notação

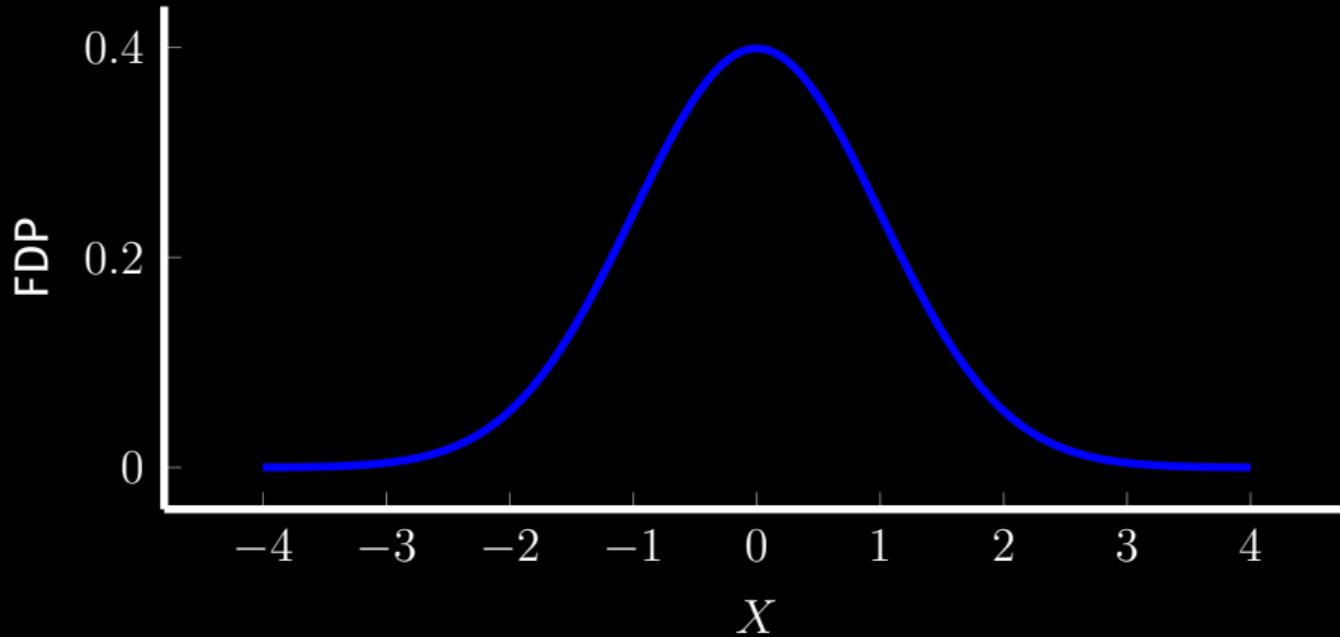
$$X \sim \text{Dist}(\theta_1, \theta_2, \dots)$$

Onde:

- $X$ : variável
- Dist: é o nome da distribuição
- $\theta_1, \theta_2, \dots$ : os parâmetros que definem como a distribuição se comporta.

Toda distribuição probabilística pode ser "parameterizada" ao especificarmos parâmetros que permitem moldarmos alguns aspectos da distribuição para algum fim específico.

# Função Distribuição de Probabilidade



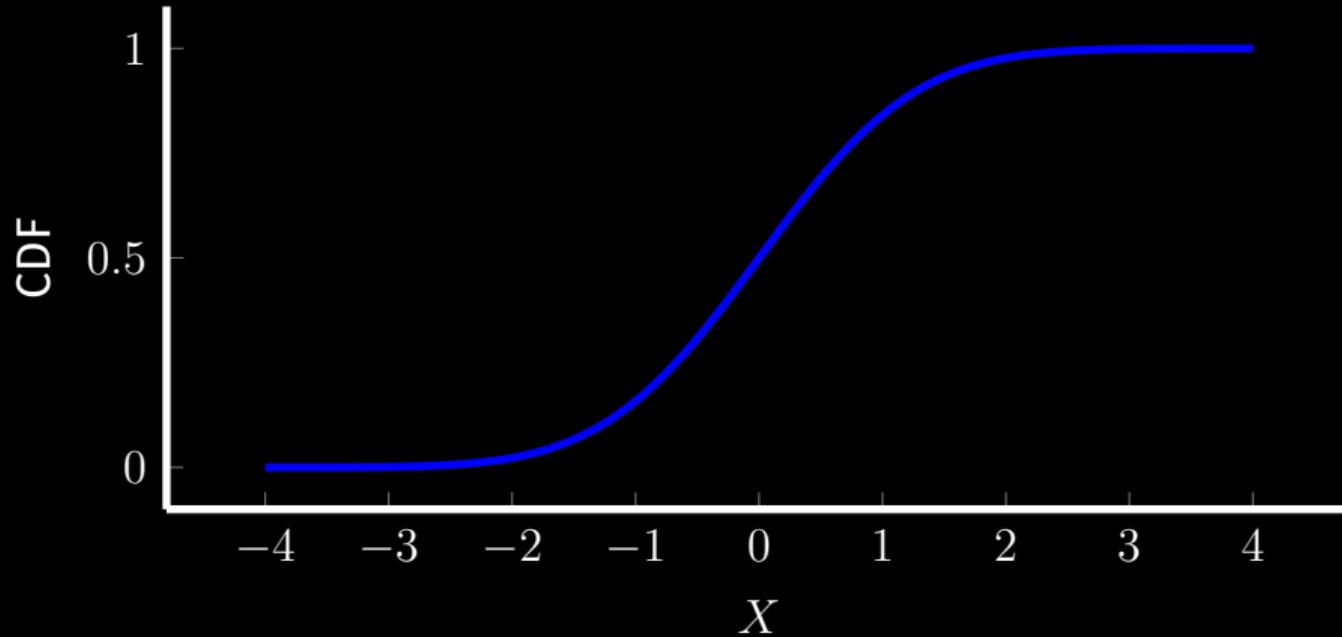
# Distribuições Probabilísticas

## Definição (Função de Distribuição Cumulativa)

A função de distribuição cumulativa (*cumulative distribution function - CDF*) de uma variável aleatória  $X$  avaliada em  $x$  é a probabilidade que  $X$  tomará valores menores ou iguais à  $x$ :

$$CDF = P(X \leq x)$$

# Função de Distribuição Cumulativa



# Distribuições Discretas

## Definição (Distribuição de Probabilidade Discreta)

*Distribuições de probabilidade discretas são aquelas que os resultados são números discretos:  $-N, \dots, -2, 1, 0, 1, 2, \dots, N$  e  $N \in \mathbb{Z}$ . Em distribuições discretas chamamos a probabilidade de uma distribuição tomar certos valores como "massa". A função massa de probabilidade FMP é a função que especifica a probabilidade da variável aleatória  $X$  tomar o valor  $x$ :*

$$FMP(x) = P(X = x)$$

# Uniforme Discreta

A distribuição uniforme discreta é uma distribuição de probabilidade simétrica em que um número finito de valores são igualmente prováveis de serem observados. Cada um dos  $n$  valores tem probabilidade igual  $\frac{1}{n}$ .

A distribuição uniforme discreta possui dois parâmetros e sua notação é  $\text{Unif}(a, b)$ :

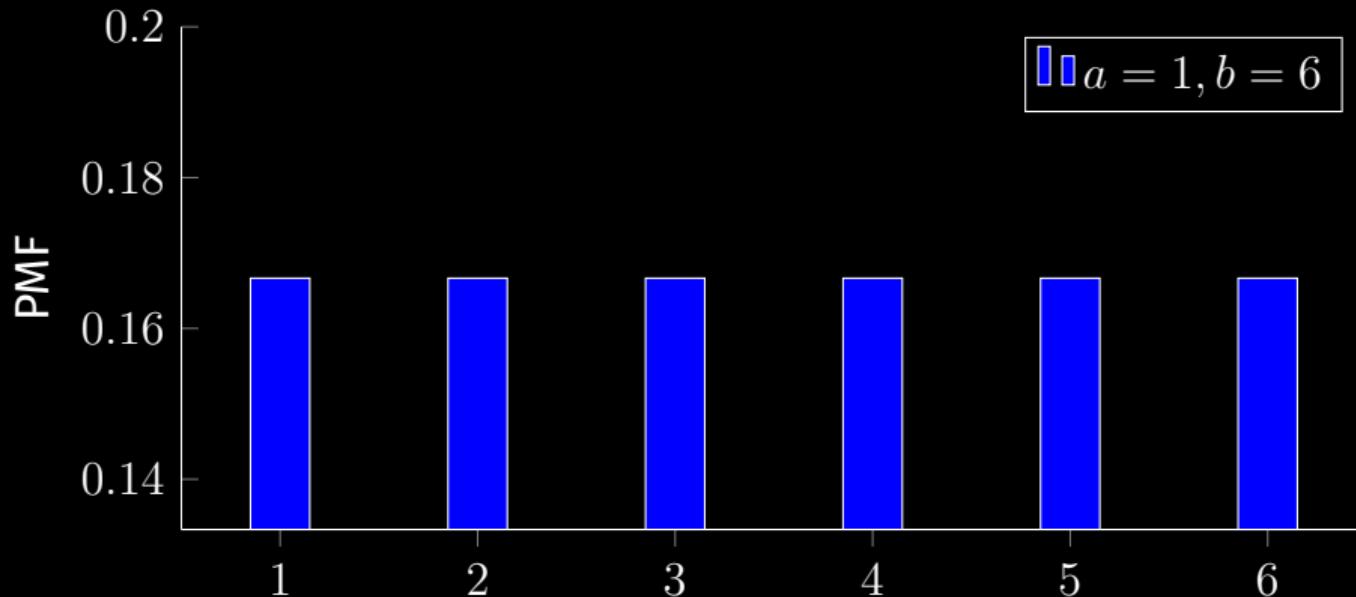
- Limite Inferior ( $a$ )
- Limite Superior ( $b$ )

Exemplo: Um dado.

# Uniforme Discreta

$$\text{Unif}(a, b) = f(x, a, b) = \frac{1}{b - a + 1} \quad \text{para } a \leq x \leq b \text{ e } x \in \{a, a + 1, \dots, b - 1, b\}$$

# Uniforme Discreta



# Bernoulli

A distribuição de Bernoulli descreve um evento binário de um sucesso de um experimento. Geralmente representamos 0 como falha e 1 como sucesso, então o resultado de uma distribuição de Bernoulli é uma variável binária  $Y \in \{0, 1\}$ .

A distribuição de Bernoulli é muito usada para modelar resultados discretos binários no qual só há dois possíveis resultados.

A distribuição de Bernoulli possui apenas um único parâmetro e sua notação é  $\text{Bernoulli}(p)$ :

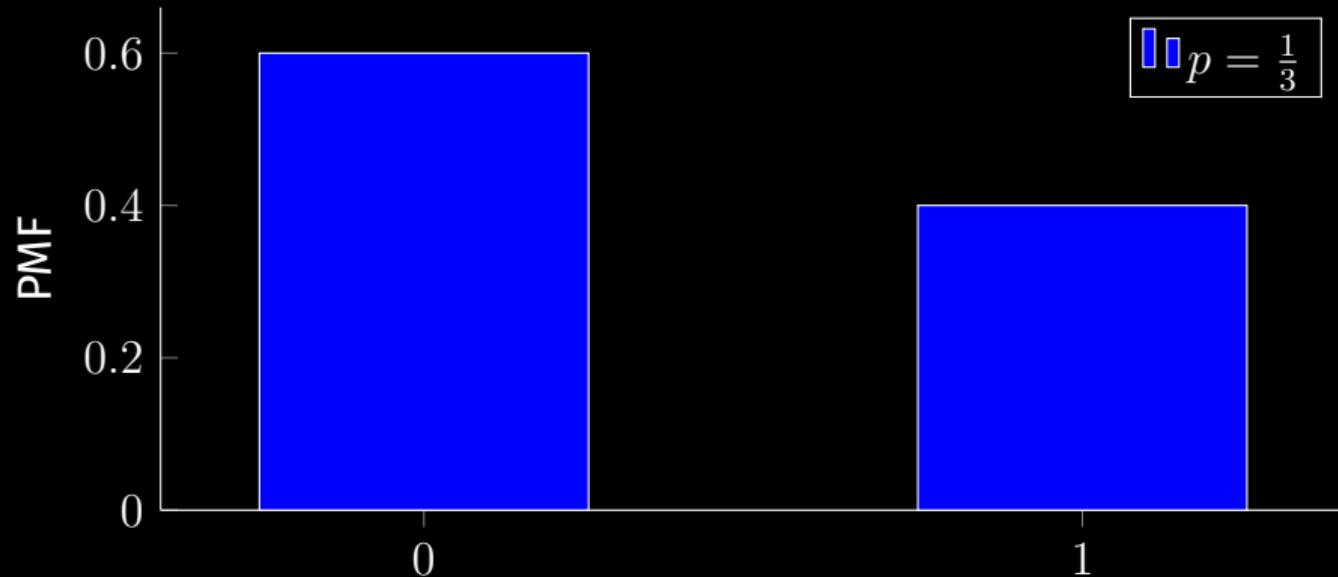
- Probabilidade de Sucesso ( $p$ )

Exemplo: Se o paciente sobreviveu ou morreu ou se o cliente conclui sua compra ou não.

# Bernoulli

**Bernoulli**( $p$ ) =  $f(x, p) = p^x(1 - p)^{1-x}$  para  $x \in \{0, 1\}$

# Bernoulli



# Binomial

A distribuição binomial descreve um evento do número de sucessos em uma sequência de  $n$  experimentos independentes, cada um fazendo uma pergunta sim-não com probabilidade de sucesso  $p$ . Note que a distribuição de Bernoulli é um caso especial da distribuição binomial no qual o número de experimentos é 1.

A distribuição binomial possui dois parâmetros e sua notação é  $\text{Bin}(n, p)$  ou  $\text{Binomial}(n, p)$ :

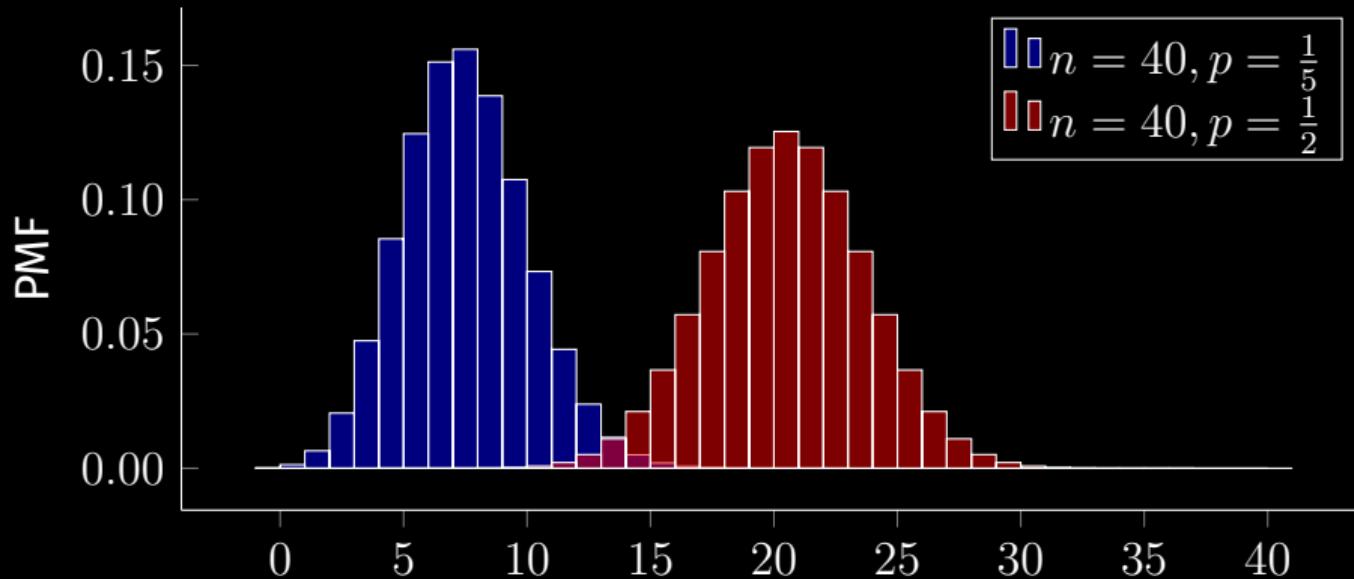
- Número de Experimentos ( $n$ )
- Probabilidade de Sucessos ( $p$ )

Exemplo: quantidade de caras em 5 lançamentos de uma moeda.

# Binomial

$$\text{Binomial}(n, p) = f(x, n, p) = \binom{n}{x} p^x (1-p)^{n-x} \quad \text{para } x \in \{0, 1, \dots, n\}$$

# Binomial



# Poisson

A distribuição Poisson expressa a probabilidade de um determinado número de eventos ocorrerem em um intervalo fixo de tempo ou espaço se esses eventos ocorrerem com uma taxa média constante conhecida e independentemente do tempo desde o último evento. A distribuição de Poisson também pode ser usada para o número de eventos em outros intervalos especificados, como distância, área ou volume.

A distribuição Poisson possui um parâmetro e sua notação é  $\text{Poisson}(\lambda)$ :

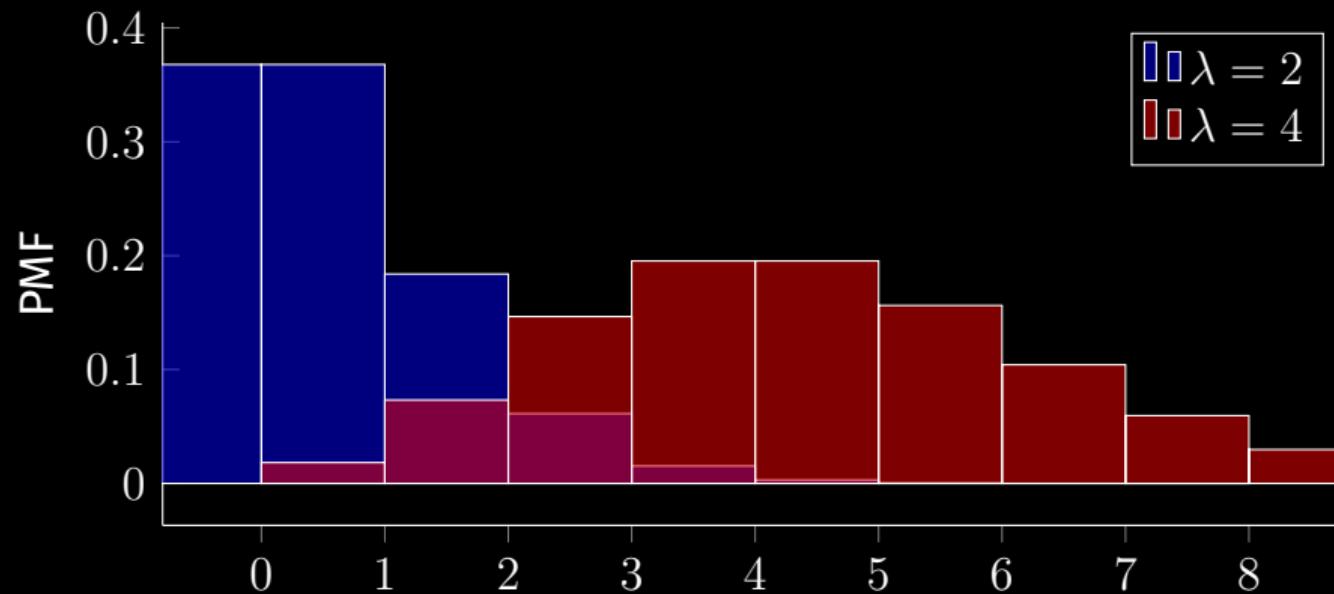
- Taxa ( $\lambda$ )

Exemplo: Quantidade de e-mails que você recebe diariamente.  
Quantidade de buracos que você encontra na rua.

# Poisson

$$\text{Poisson}(\lambda) = f(x, \lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \quad \text{para } \lambda > 0$$

# Poisson



# Binomial Negativa<sup>24</sup>

A distribuição binomial negativa descreve um evento do número de sucessos em uma sequência de  $n$  experimentos independentes, cada um fazendo uma pergunta sim-não com probabilidade  $p$  até que se obtenha  $k$  sucessos. Note que ela se torna idêntica à distribuição de Poisson quando no limite de  $k \rightarrow \infty$ . Isto faz com que seja uma opção robusta para substituir uma distribuição de Poisson para modelar fenômenos com uma *superdispersão* (variação nos dados excedente ao esperado).

A distribuição negativa binomial possui dois parâmetros e sua notação é Binomial Negativa( $k, p$ ):

- Número de Sucessos ( $k$ )
- Probabilidade de Sucessos ( $p$ )

Exemplo: Contagem anual de ciclones tropicais.

---

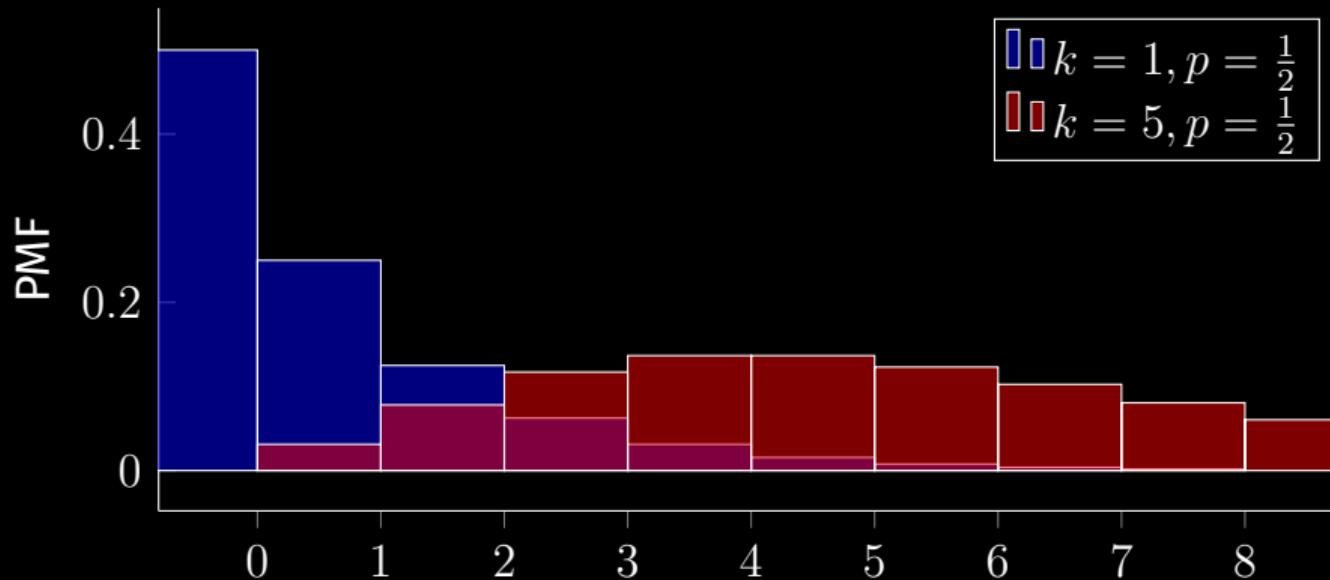
<sup>24</sup>Qualquer fenômeno que pode ser modelo com uma distribuição de Poisson, pode ser modelo com uma distribuição binomial negativa (Gelman et al., 2013b; Gelman, Hill & Vehtari, 2020).

# Binomial Negativa

$$\text{Binomial Negativa}(k, p) = f(x, k, p) = \binom{x + k - 1}{k - 1} p^x (1 - p)^k$$

para  $x \in \{0, 1, \dots, n\}$

# Binomial Negativa



# Distribuições Contínuas

## Definição (Distribuição de Probabilidade Contínua)

*Distribuições de probabilidade contínuas são aquelas que os resultados são valores em uma faixa contínua (também chamados de número reais):  $(-\infty, +\infty) \in \mathbb{R}$ . Em distribuições contínuas chamamos a probabilidade de uma distribuição tomar certos valores como "densidade". Como estamos falando sobre números reais não conseguimos obter a probabilidade de uma variável aleatória  $X$  tomar o valor de  $x$ . Isto sempre será 0, pois não há como especificar um valor exato de  $x$ .  $x$  vive na linha dos números reais, portanto, precisamos especificar a probabilidade de  $X$  tomar valores em um **intervalo**  $[a, b]$ . A função densidade de probabilidade FDP é definida como:*

$$FDP(x) = P(a \leq X \leq b) = \int_a^b f(x)dx$$

# Uniforme Contínua

A distribuição uniforme contínua é uma distribuição de probabilidade simétrica em que um número infinito de intervalos de valores são igualmente prováveis de serem observados. Cada um dos  $n$  infinitos intervalos valores tem probabilidade igual  $\frac{1}{n}$ .

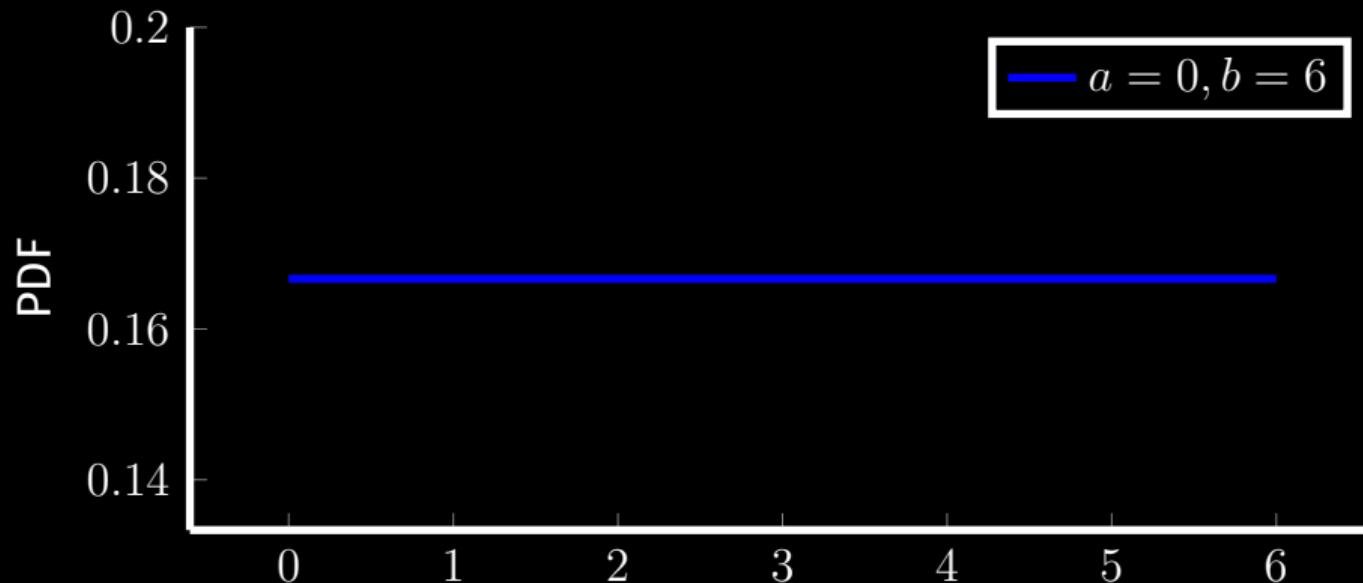
A distribuição uniforme contínua possui dois parâmetros e sua notação é  $\text{Unif}(a, b)$ :

- Limite Inferior ( $a$ )
- Limite Superior ( $b$ )

# Uniforme Contínua

$$\text{Unif}(a, b) = f(x, a, b) = \frac{1}{b - a} \quad \text{para } a \leq x \leq b \text{ e } x \in [a, b]$$

# Uniforme Contínua



## Normal

Essa distribuição geralmente é usada nas ciências sociais e naturais para representar variáveis contínuas na qual as suas distribuições não são conhecidas. Esse pressuposto é por conta do teorema do limite central. O teorema do limite central afirma que, em algumas condições, a média de muitas amostras (observações) de uma variável aleatória com média e variância finitas é ela própria uma variável aleatória cuja distribuição converge para uma distribuição normal à medida que o número de amostras aumenta.

Portanto, as quantidades físicas que se espera sejam a soma de muitos processos independentes (como erros de medição) muitas vezes têm distribuições que são quase normais.

# Normal

A distribuição normal possui dois parâmetros e sua notação é  $\text{Normal}(\mu, \sigma^2)$  ou  $N(\mu, \sigma^2)$ :

- Média ( $\mu$ ): média da distribuição e também a moda e a mediana
- Desvio Padrão ( $\sigma$ ) (às vezes também parametrizada com a variância  $\sigma^2$ ): é uma medida de dispersão das observações em relação à média

Exemplo: Altura, Peso etc.

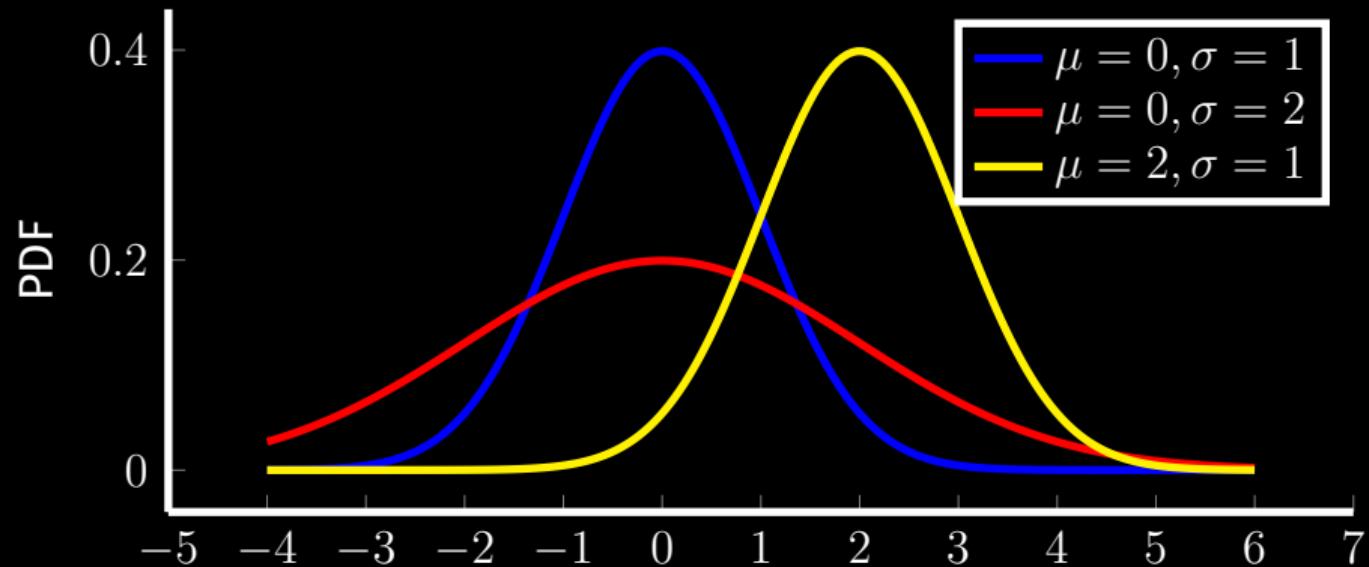
# Normal<sup>25</sup>

$$\text{Normal}(\mu, \sigma) = f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{para } \sigma > 0$$

---

<sup>25</sup>veja como a distribuição Normal foi derivada a partir da distribuição binomial nos Slides de Backup no final dessa apresentação

# Normal



# Log-Normal

A distribuição Log-normal é uma distribuição de probabilidade contínua de uma variável aleatória cujo logaritmo é normalmente distribuído. Assim, se a variável aleatória  $X$  for distribuída normalmente por log natural ( $\ln$ ), então  $Y = \ln(X)$  terá uma distribuição normal.

Uma variável aleatória com distribuição logarítmica aceita apenas valores reais positivos. É um modelo conveniente e útil para medições em ciências exatas e de engenharia, bem como medicina, economia e outros campos, por ex. para energias, concentrações, comprimentos, retornos financeiros e outros valores.

Um processo log-normal é a realização estatística do produto multiplicativo de muitas variáveis aleatórias independentes, cada uma das quais positiva.

# Log-Normal

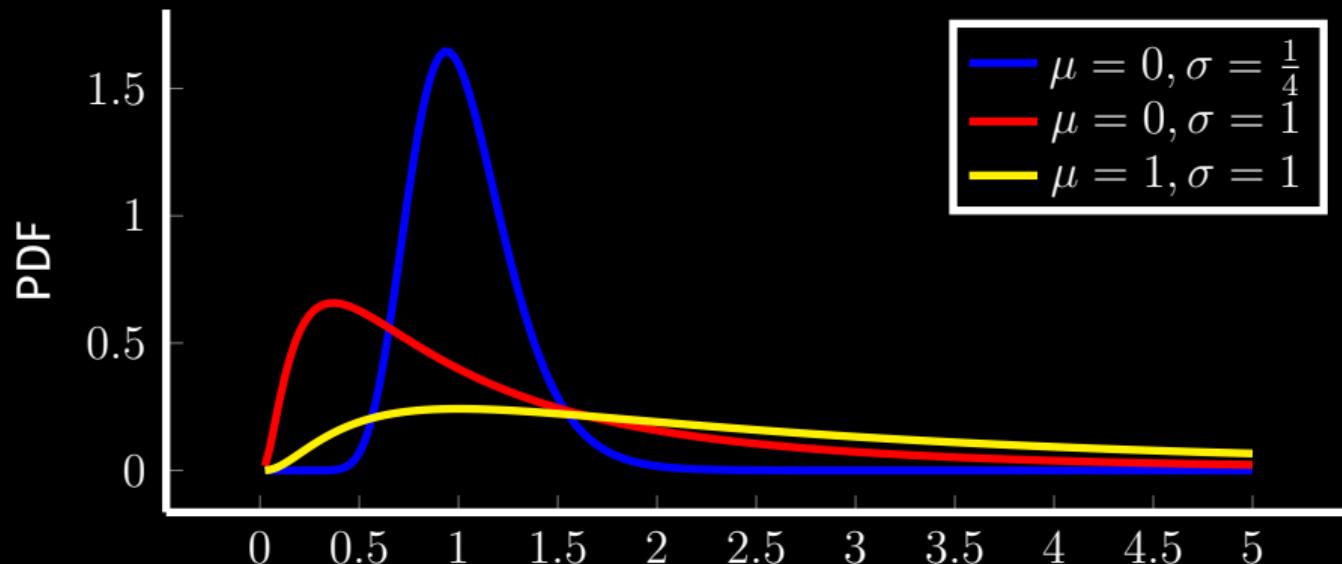
A distribuição log-normal possui dois parâmetros e sua notação é Log-Normal( $\mu, \sigma^2$ ):

- Média ( $\mu$ ): média do logaritmo natural ( $\ln$ ) da distribuição
- Desvio Padrão ( $\sigma$ ): a variância do logaritmo natural da distribuição ( $\sigma^2$ ) é uma medida de dispersão das observações em relação à média

# Log-Normal

$$\text{Log-Normal}(\mu, \sigma) = f(x, \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\left(\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right)} \quad \text{para } \sigma > 0$$

# Log-Normal



# Exponencial

A distribuição exponencial é a distribuição de probabilidade do tempo entre eventos que ocorrem de forma contínua e independente a uma taxa média constante.

A distribuição exponencial possui um parâmetro e sua notação é  $\text{Exp}(\lambda)$ :

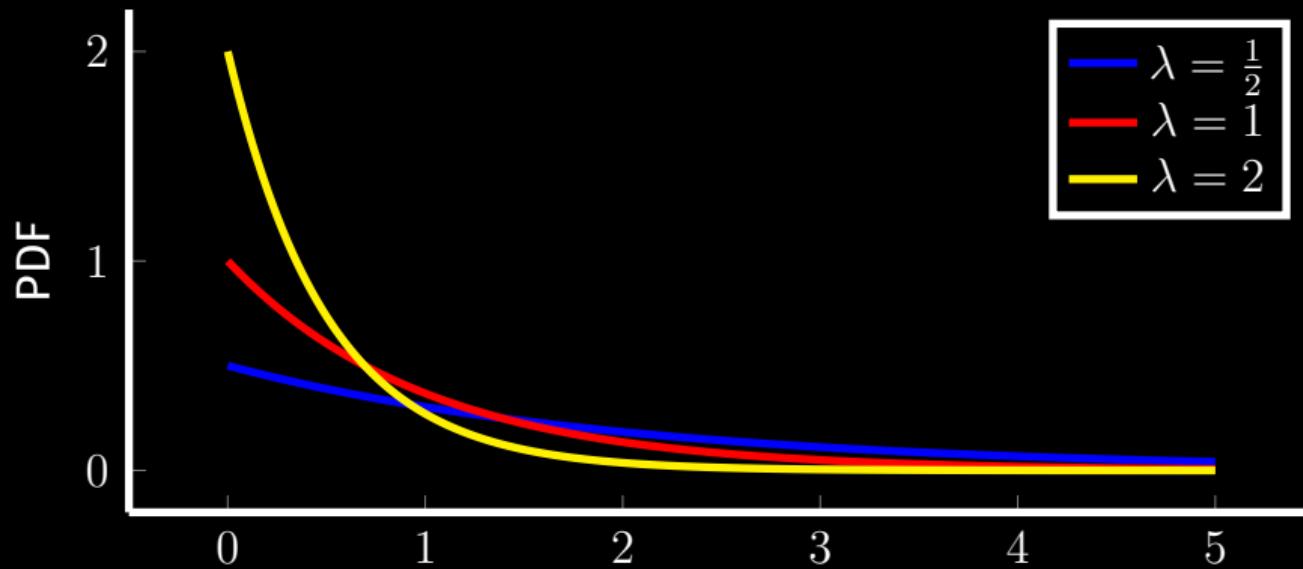
- Taxa ( $\lambda$ )

Exemplo: Quanto tempo até o próximo terremoto. Quanto tempo até o próximo ônibus.

# Exponencial

$$\text{Exp}(\lambda) = f(x, \lambda) = \lambda e^{-\lambda x} \quad \text{para } \lambda > 0$$

# Exponencial



# *t* de Student

A distribuição *t* de Student surge ao estimar a média de uma população normalmente distribuída em situações onde o tamanho da amostra é pequeno e o desvio padrão da população é desconhecido<sup>26</sup>.

Se tomarmos uma amostra de  $n$  observações de uma distribuição normal, então a distribuição *t* com  $\nu = n - 1$  graus de liberdade pode ser definida como a distribuição da localização da média da amostra em relação à média verdadeira, dividida pelo desvio padrão da amostra, após multiplicar pelo termo padronizador  $\sqrt{n}$ .

A distribuição *t* é simétrica e em forma de sino, como a distribuição normal, mas tem caudas mais longas, o que significa que é mais propensa a produzir valores que estão longe de sua média.

---

<sup>26</sup> Daqui que vem o tal do teste *t* de Student

# *t* de Student

A distribuição *t* de Student possui um parâmetro e sua notação é  $\text{Student}(\nu)$ :

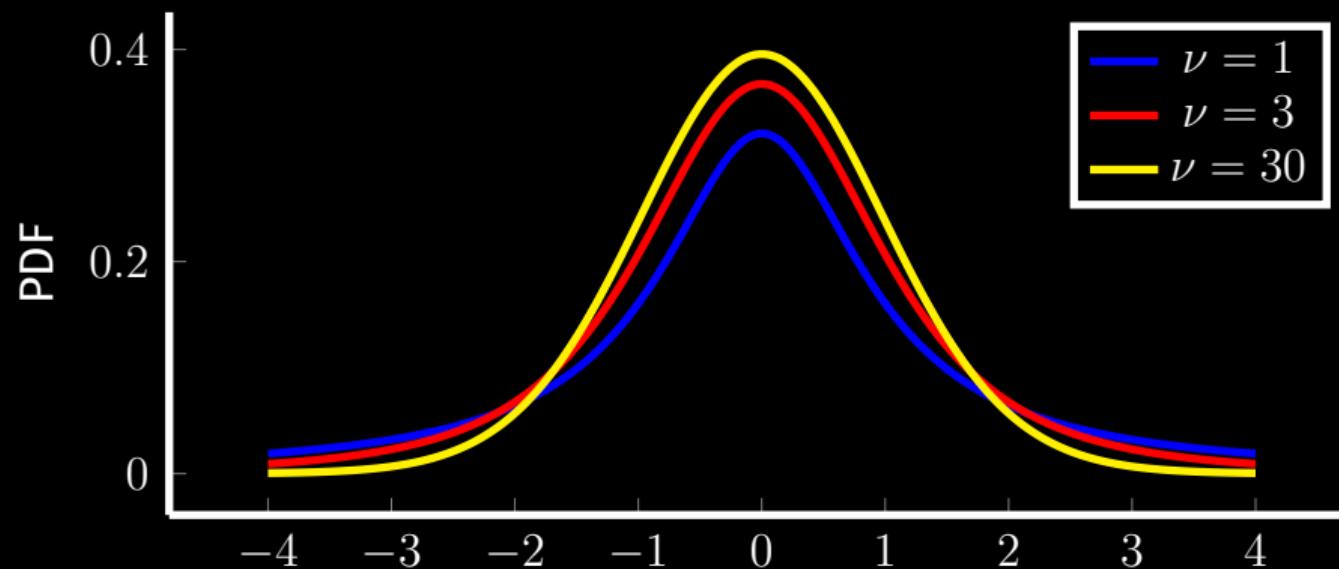
- Graus de Liberdade ( $\nu$ ): controla o quanto ela se assemelha com uma distribuição normal

Exemplo: Uma base de dados cheia de outliers.

# *t* de Student

$$\text{Student}(\nu) = f(x, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \quad \text{para } \nu \geq 1$$

# *t* de Student



# Beta

A distribuição beta é uma escolha natural para modelar qualquer coisa que seja restrita a valores entre 0 e 1. Portanto, é uma boa candidata para probabilidades e proporções.

A distribuição beta possui dois parâmetros e sua notação é Beta( $\alpha, \beta$ ):

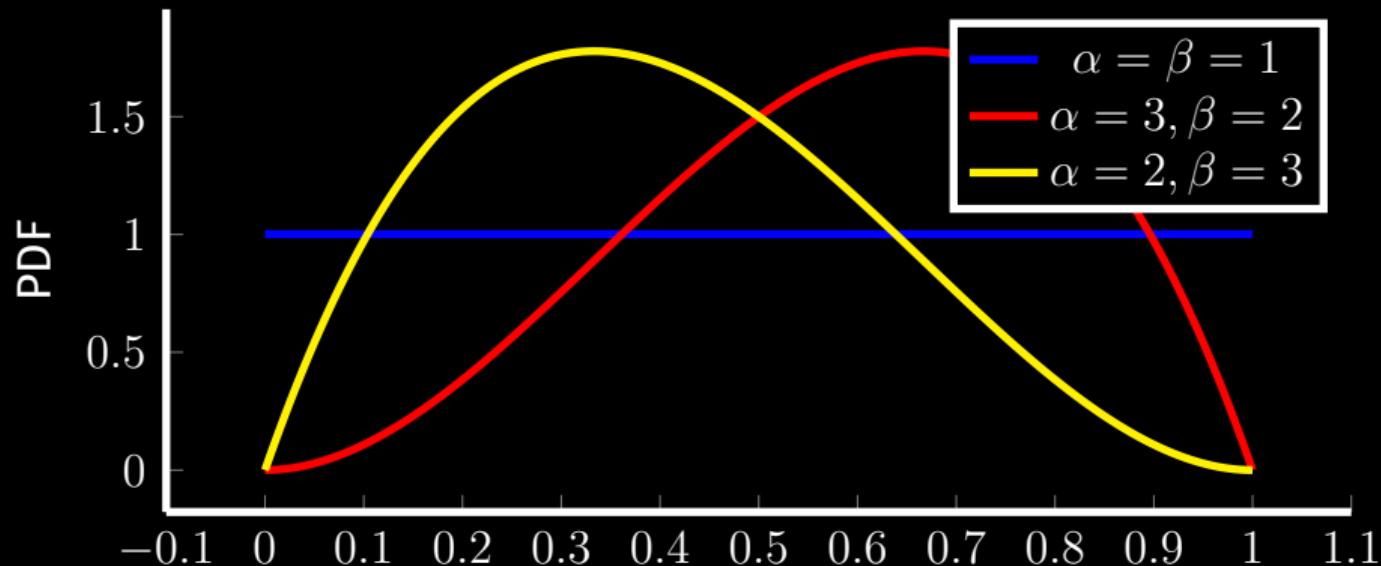
- Parâmetro de Forma ( $\alpha$  ou às vezes  $a$ ): controla o quanto a forma é deslocada para próximo de 1
- Parâmetro de Forma ( $\beta$  ou às vezes  $b$ ): controla o quanto a forma é deslocada para próximo de 0

Exemplo: Um jogador de basquete já marcou 5 lances livres e errou 3 em um total de 8 tentativas - Beta(3, 5)

# Beta

$$\text{Beta}(\alpha, \beta) = f(x, \alpha, \beta) \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}} \quad \text{para } \alpha, \beta > 0 \text{ e } x \in [0, 1]$$

# Beta



# Sumário para rstanarm e brms

## 3.1 Leituras Recomendadas

## 3.2 Ecosistema Stan

### 3.2.1 Interfaces Oficiais

### 3.2.2 Pacotes

## 3.3 Especificação de Modelos com Fórmulas

## 3.4 Funções de Verossimilhança com family

## 3.5 rstanarm

### 3.5.1 Modelos do rstanarm

### 3.5.2 Visualizações do rstanarm

### 3.5.3 Otimizações do rstanarm

## 3.6 brms

### 3.6.1 Modelos do brms

### 3.6.2 Visualizações do brms

### 3.6.3 Otimizações do brms

# rstanarm e brms - Leituras Recomendadas<sup>27</sup>

- Vinheta e Manual do rstanarm de Goodrich et al. (2020)
- Vinheta e Manual do brms de Bürkner (2017)
- Tutorial de rstanarm de Muth et al. (2018)
- Tutorial de brms de Bürkner (2018)
- *Workflow Bayesiano* de Gelman, Vehtari et al. (2020)
- *Workflow Visual Bayesiano* de Gabry et al. (2019)
- Vinheta e Manual do bayesplot Gabry e Mahr (2021)
- Storopoli (2021) - rstanarm e brms

---

<sup>27</sup> rstanarm - <http://mc-stan.org/rstanarm/> e brms - <https://paul-buerkner.github.io/brms/>

# O que é Stan

Stan (Carpenter et al., 2017) é uma **plataforma para modelagem e computação estatística de alto desempenho**. Milhares de usuários contam com Stan para modelagem estatística, análise de dados e previsão nas ciências sociais, biológicas e físicas, engenharia e negócios.



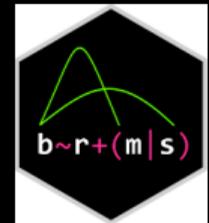
# Interfaces do Stan

- R: RStan e CmdStanR
- Python: PyStan e CmdStanPy
- Shell (Linha de Comando): CmdStan
- Julia: Stan.jl
- Scala: ScalaStan
- Matlab: MatlabStan
- Stata: StataStan
- Mathematica: MathematicaStan

# Interfaces Amigáveis do Stan

Somente para R

- `rstanarm` (Goodrich et al., 2020): ajuda o usuário a especificar modelos usando a sintaxe familiar de fórmulas do R.
- `brms` (Bürkner, 2017): similar ao `rstanarm` pois usa a sintaxe familiar de fórmulas do R, mas dá maior flexibilidade na especificação de modelos mais complexos.



# Código Stan

```
data {
  int<lower=0> N;
  vector[N] x1;
  vector[N] x2;
  vector[N] y;
}
parameters {
  real alpha;
  vector[2] beta;
  real<lower=0> sigma;
}
model {
  sigma ~ cauchy(0, 2.5);
  y ~ normal(alpha + beta[1] * x1 + beta[2] * x2, sigma);
}
```

## rstanarm versus brms

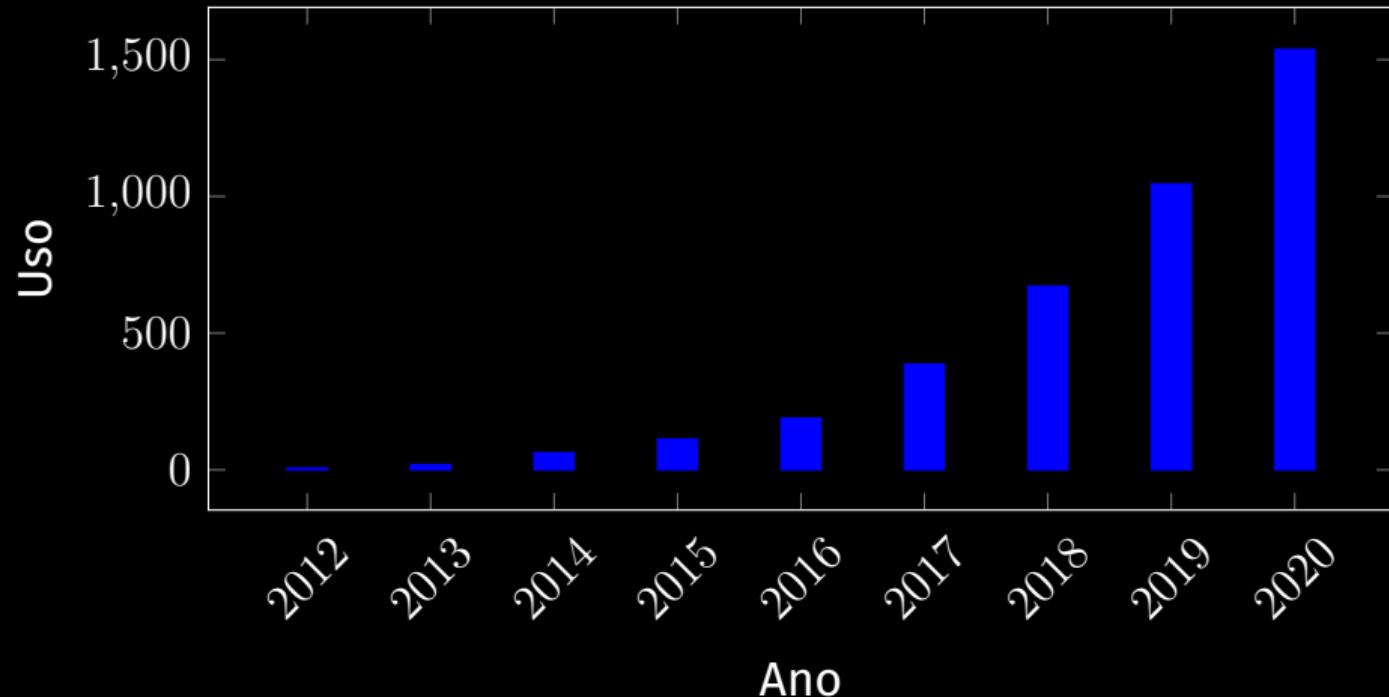
Para remediar essa barreira de acesso ao Stan, temos interfaces abstratas que interpretam a intenção do usuário e lidam com a parte mais *obral* de codificação:

- rstanarm todos os modelos são pré-compilados e brms não possui os modelos pré-compilados então os modelos devem ser todos compilados antes de serem rodados
- Como brms compila os modelos conforme a especificação do usuário, ele pode criar modelos um pouco mais eficientes que o rstanarm
- brms dá maior poder e flexibilidade ao usuário na especificação de funções de verossimilhança e também permite modelos mais complexos que o rstanarm

# Pacotes R

- bayesplot – visualização
- loo – seleção de modelos com LOO-CV
- projpred – seleção de variáveis
- prophet – previsão de séries temporais
- varstan – modelos *Vector Auto-Regressive* (VAR)
- blavaan – *Structural Equation Modeling* (SEM)
- lgpr – *Longitudinal Gaussian Process Regression* (LGPR)
- bayesforecast – previsão de séries temporais
- baggr – meta-análise
- CausalQueries – modelos binários causais

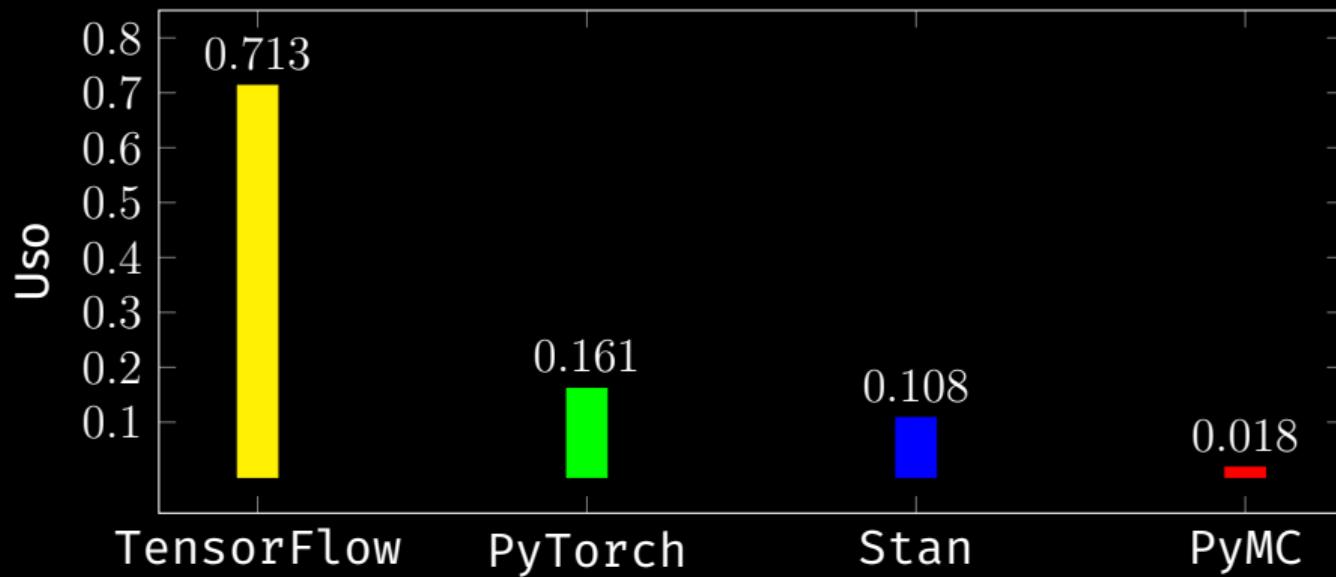
# Stan<sup>28</sup> na Scopus<sup>29</sup>



<sup>28</sup>foi lançado em 2012

<sup>29</sup>veja as buscas Scopus nos Slides de Backup no final dessa apresentação

Stan<sup>3031</sup>



<sup>30</sup>baseado no reporte anual do Breck Baldwin para a NUMFocus

<sup>31</sup>veja as buscas Scopus nos Slides de Backup no final dessa apresentação

# Especificação de Modelos com Fórmulas

Todos os modelos especificados pelo `rstanarm` e `brms` usam uma fórmula com a seguinte sintaxe:

```
dependente ~ independente_1 + independente_2 + ...
```

# Especificação de Modelos com Fórmulas

Moderações?! Sem problema:

```
dependente ~ independente_1 * moderadora + independente_2 * moderadora + ...
```

# Funções de Verossimilhança com family

Todo modelo especificado pelo `rstanarm` e `brms` devem especificar qual família da função de verossimilhança (`family`) respectivamente com a função de ligação (`link`) que fará o mapeamento dos parâmetros condicionados nos dados para a variável dependente<sup>32</sup>.

---

<sup>32</sup>caso o usuário não designe nenhum valor para esses dois parâmetros, `rstanarm` e `brms` usarão a verossimilhança Gaussiana (`family = gaussian`) e a função de identidade como função de ligação (`link = "identity"`)

# Funções de Verossimilhança com family

- Gaussiana – family = gaussian(link = "identity")
- Log-Normal – family = lognormal(link = "log")
- Binomial – family = binomial(link = "logit")
- Poisson – family = poisson(link = "log")
- Binomial Negativa – family = negbinomial(link = "log")
- t de Student – family = student(link = "identity")
- Exponencial – family = exponential(link = "log")

## rstanarm<sup>33</sup>

O rstanarm é a porta de entrada para estatística Bayesiana com Stan.

O nome rstanarm é:

- r: pacote para R
- stan: usa a linguagem probabilística Stan
- arm: acrônimo para *Applied Regression Modeling*

---

<sup>33</sup>Goodrich et al. (2020)

# Modelos do `rstanarm`<sup>34</sup>

- `stan_glm()` – modelos lineares generalizados (*generalized linear model*)
- `stan_lm()` – modelos lineares regularizados (*linear model*)
- `stan_aov()` – modelo ANOVA (*analysis of variance*)
- `stan_glmer()` - modelos linares generalizados multiníveis
- `stan_lmer()` – modelos linares regularizados multiníveis
- `stan_jm()` – modelos longitudinais e de sobrevivência
- `stan_nlmer()` – modelos não-lineares multiníveis (*non-linear model*)
- `stan_polr()` – modelos ordinais
- `stan_gamm4()` – modelos aditivos linares multiníveis

<sup>34</sup>Neste curso usaremos apenas `stan_glm` e `stan_glmer`, mas saiba que você possui uma vasta categoria de modelos Bayesianos à disposição

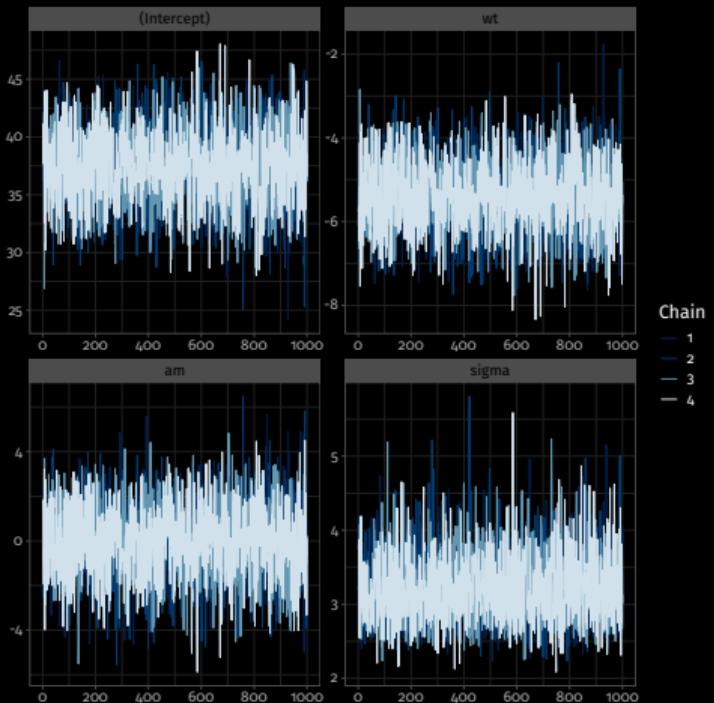
# Exemplo Simples do rstanarm

```
library(rstanarm)
rstanarm_fit <- stan_glm(mpg ~ wt + am, data = mtcars)
summary(rstanarm_fit)
```

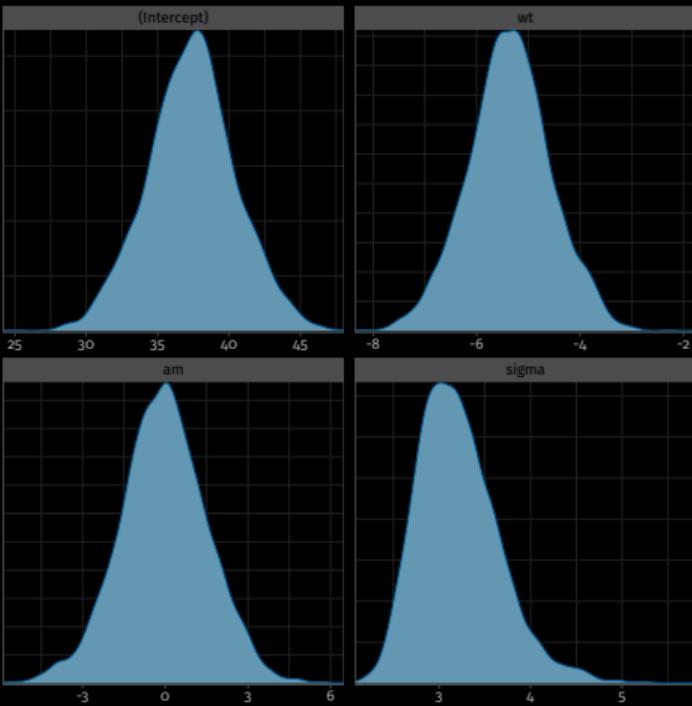
# summary do rstanarm

Parameter	Rhat	n_eff	mean	sd	2.5%	50%	97.5%
(Intercept)	1.00	2157	37.16	3.20	30.78	37.16	43.44
wt	1.00	2285	-5.32	0.83	-6.91	-5.32	-3.65
am	1.00	2263	0.07	1.62	-3.11	0.07	3.15

# Visualizações do rstanarm



# Visualizações do rstanarm



# Otimizações do rstanarm

- Por padrão rstanarm usa **NCP**<sup>35</sup> em **modelos multiníveis**<sup>36</sup>
- Por padrão rstanarm usa **centraliza as covariáveis em zero** para facilitar o trabalho do amostrador MCMC.
- Em modelos binomiais (regressão logística) é mais eficiente computacionalmente especificar uma fórmula usando o **cbind** –  
`stan_glm(cbind(successos, n - successos) ~ ...)`
- É possível especificar para que rstanarm use uma decomposição QR<sup>37</sup> da matrix de dados facilitando o trabalho do amostrador MCMC  
`- stan_glm(..., QR = TRUE)`

---

<sup>35</sup>*Non-Centered Parameterization*

<sup>36</sup>mais sobre isso quando falamos de modelos multiníveis

<sup>37</sup>veja os detalhes matemáticos nos Slides de Backup no final dessa apresentação

# brms<sup>38</sup>

O brms alia toda a comodidade do rstanarm com o poder e flexibilidade do Stan. O nome brms quer dizer:

- b: *bayesian*
- r: *regression*
- m: *models*
- s: usando a linguagem probabilística Stan

---

<sup>38</sup>Bürkner (2017)

# Modelos do brms

Ao invés de possuir diversas funções para diferentes tipos de modelo, brms tem apenas uma única função para especificar modelos – brm (...) O usuário consegue especificar qualquer modelo que quiser a partir da função brm (...) apenas mudando seus parâmetros internos:

- **family** – especifica a família da função de verossimilhança do modelo (padrão **gaussian**)
- **link** – especifica a função de ligação que fará o mapeamento dos parâmetros condicionados nos dados para a variável dependente do modelo (padrão varia conforme o **family**)

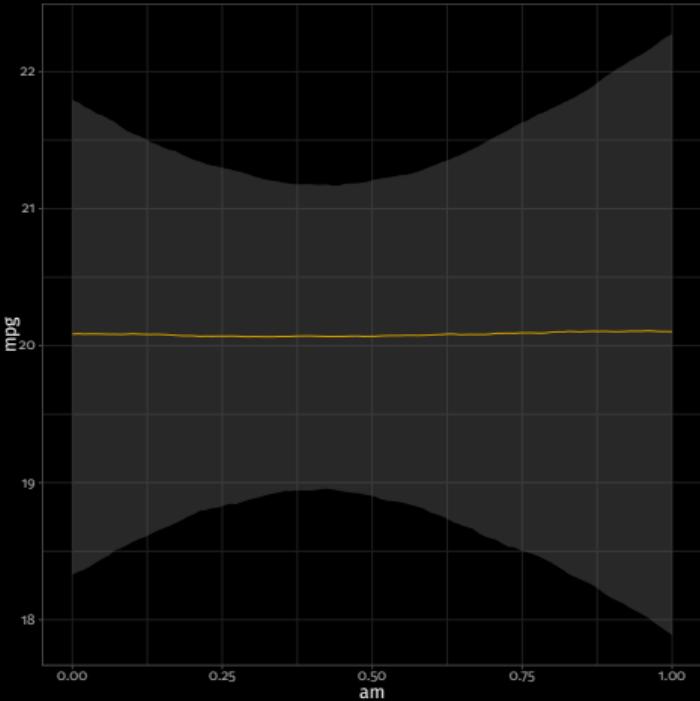
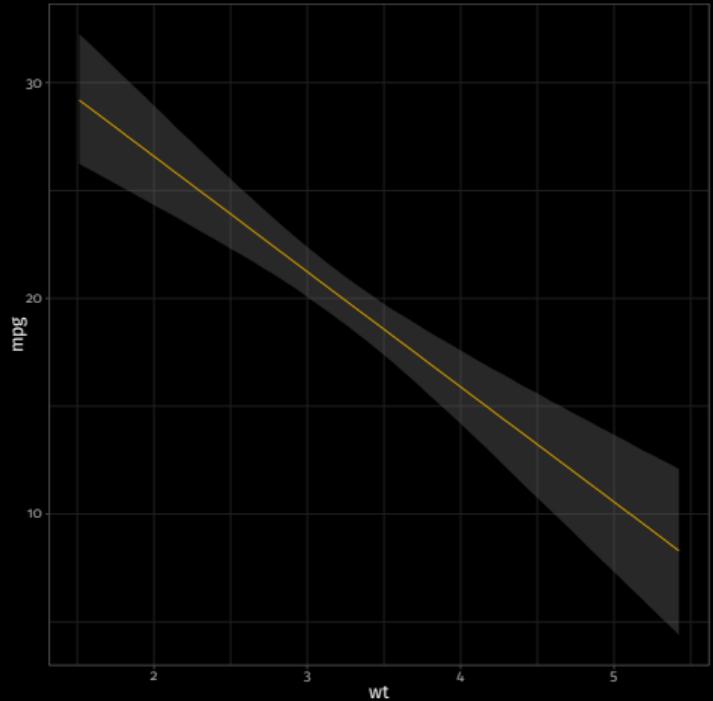
# Exemplo Simples do brms

```
library(brms)
brms_fit <- brm(mpg ~ wt + am, data = mtcars)
summary(brms_fit)
```

## summary do brms

Parameter	Rhat	n_eff	mean	sd	2.5%	50%	97.5%
(Intercept)	1.00	2361	37.28	3.22	30.83	37.30	43.79
wt	1.00	2486	-5.35	0.83	-7.03	-5.35	-3.71
am	1.00	2557	-0.00	1.61	-3.26	0.02	3.28

# Visualizações do brms



# Otimizações do brms

- Por padrão brms **também** usa **NCP<sup>39</sup>** em **modelos multiníveis<sup>40</sup>**
- Por padrão brms **também** usa **centraliza as covariáveis em zero** para facilitar o trabalho do amostrador MCMC. Mas, você consegue desativar com `bf(formula, center = FALSE)` sendo input da fórmula do brms
- Em modelos binomiais (regressão logística) é mais eficiente computacionalmente especificar uma fórmula usando o `trials` – `brms(success | trials(n) ~ ...)`
- É possível especificar para que `rstanarm` use uma decomposição QR<sup>41</sup> da matrix de dados facilitando o trabalho do amostrador MCMC – `bf(formula, decomp = "QR")` sendo input da fórmula do brms

<sup>39</sup>Non-Centered Parameterization

<sup>40</sup>mais sobre isso quando falamos de modelos multiníveis

<sup>41</sup>veja os detalhes matemáticos nos Slides de Backup no final dessa apresentação

## *Backend CmdStanR com o brms*

Além disso, brms permite o uso de outros *backend* além do RStan.

Em especial o CmdStanR sempre está mais atualizado que o RStan e tem algumas otimizações interessantes. Uma delas é o normalize que usa o log PDF **não-normalizado** da função de verossimilhança, assim evitando computação de constantes ao usar o log PDF **normalizado** da função de verossimilhança<sup>42</sup>:

```
brm(..., backend = "cmdstanr", normalize = FALSE)
```

---

<sup>42</sup>eu vejo, dependendo do modelo, um ganho de 25% de velocidade

# Sumário para *Prioris*

4.1 Leituras Recomendadas

4.2 A subjetividade da *Priori*

4.3 Tipos de *Prioris*

4.3.1 *Priori Uniforme (Flat)*

4.3.2 *Priori Fracamente Informativa (Weekly Informative)*

4.3.3 *Priori Informativa (Informative)*

4.4 *Prioris* do rstanarm e brms

4.4.1 *Prioris* do rstanarm

4.4.2 *Prioris* do brms

# Prioris - Leituras Recomendadas

- Gelman et al. (2013b):
  - Capítulo 2: Single-parameter models
  - Capítulo 3: Introduction to multiparameter models
- McElreath (2020) - Capítulo 4: Geocentric Models
- Gelman, Hill e Vehtari (2020):
  - Capítulo 9, Seção 9.3: Prior information and Bayesian synthesis
  - Capítulo 9, Seção 9.5: Uniform, weakly informative, and informative priors in regression
- van de Schoot et al. (2021)
- Storopoli (2021) - Priors
- Vinheta do `rstanarm` sobre *prioris*
- Documentação sobre *prioris* do `brms`

# Probabilidade *Priori*

A Estatística Bayesiana é caracterizada pelo uso de informação prévia embutida como probabilidade prévia  $P(\theta)$ , chamada de *priori*:

$$\underbrace{P(\theta \mid y)}_{Posterior} = \frac{\overbrace{P(y \mid \theta) \cdot P(\theta)}^{\text{Verossimilhança } Prior}}{\underbrace{P(y)}_{\text{Constante Normalizadora}}}$$

# A subjetividade da *Priori*

- Muitas críticas à estatística Bayesiana, se dá pela subjetividade da elucidação da probabilidade *\*a priori\** de certas hipóteses ou parâmetros de modelos.
- A subjetividade é algo indesejado na idealização do cientista e do método científico.
- Tudo que envolve ação humana nunca será 100% objetivo. Temos subjetividade em tudo e ciência **não** é um exceção.
- O próprio processo dedutivo e criativo de formulação de teoria e hipóteses não é algo objetivo.
- A estatística frequentista, que bane o uso de probabilidades *a priori* também é subjetiva, pois há **MUITA** subjetividade em especificar um modelo e uma função de verossimilhança (Jaynes, 2003; van de Schoot et al., 2021)

# Como Incorporar Subjetividade

- A estatística Bayesiana **abraça** a subjetividade enquanto a estatística frequentista a **proíbe**.
- Para a estatística Bayesiana, **subjetividade guiam nossas inferências e nos levam a modelos mais robustos**, confiáveis e que podem auxiliar à tomada de decisão.
- Já para a estatística frequentista, **subjetividade é um tabu e todas as inferências devem ser objetivas**, mesmo que isso resulte em **esconder pressupostos dos modelos embaixo dos panos**.
- Estatística Bayesiana possui também pressupostos e subjetividade, mas estes são **enunciados e formalizados**.

## Tipos de *Prioris*

De maneira geral, podemos ter 3 tipos de *priori* em uma abordagem Bayesiana (Gelman et al., 2013b; McElreath, 2020; van de Schoot et al., 2021):

- uniforme (*flat*): não recomendada
- fracamente informativa (*weakly informative*): pequena restrição com um pouco de senso comum e baixo conhecimento de domínio incorporado
- informativa (*informative*): conhecimento de domínio incorporado

## Priori Uniforme (Flat)

Parte da premissa que "tudo é possível". Não há limites na crença de que tamanho o valor deve ser ou quaisquer restrições.

*Prioris* uniformes e super-vagas geralmente não são recomendadas e algum esforço deve ser incluído para ter, pelo menos, *prioris* um pouco informativas.

Formalmente uma *priori* uniforme é uma distribuição uniforme em todo o suporte possível do valor do parâmetros

- parâmetros a serem estimados:  $\{\theta \in \mathbb{R} : -\infty < \theta < \infty\}$
- resíduos ou erros do modelo:  $\{\sigma \in \mathbb{R}^+ : 0 \leq \sigma < \infty\}$

## Priori Fracamente Informativa (*Weekly Informative*)

Aqui começamos a entrar num palpite informado sobre o valor do parâmetro. Portanto não partimos da premissa que "tudo é possível".

Recomendo sempre traduzir as *prioris* do seu problema em algo centrado em 0 e com desvio padrão 1<sup>43</sup>:

- $\theta \sim \text{Normal}(0, 1)$  (preferida do Andrew Gelman<sup>44</sup>)
- $\theta \sim \text{Student}(\nu = 3, 0, 1)$  (preferida do Aki Vehtari)
- $\sigma \sim \text{Exponencial}(10)$

---

<sup>43</sup>isso chama-se normalização, transformar todas variáveis para terem média 0 e desvio padrão 1

<sup>44</sup>Veja mais sobre a escolha de *prioris* nessa wiki do GitHub do Stan

# Um exemplo de uma *Priori* robusta

Um exemplo interessante vem de uma aula do Ben Goodrich<sup>45</sup> professor de Columbia e membro do grupo de pesquisa de Stan.

Aqui ele fala sobre um dos maiores efeitos observados nas ciências sociais. Nas pesquisas de intenção de voto à eleição presidencial dos EUA de 2008 (Obama vs McCain), havia um apoio de quase 40% do Obama de maneira geral. Se você trocasse a raça de um respondente de `race_black = 0` para `race_black = 1` isso gerava um aumento de aproximadamente 60% na probabilidade do respondente votar no Obama<sup>46</sup>.

Em escala logit esses 60% se traduziriam em um modelo binomial como um coeficiente  $\beta_{Race\ Black} = 3.64^{47}$ . Esse tamanho de efeito seria facilmente inferido com uma *priori*

$$\beta_{Race\ Black} \sim \text{Normal}(0, 1)$$

---

<sup>45</sup><https://youtu.be/p6cyRBWahRA>, caso queira ver o vídeo na íntegra, a parte que nos interessa de *priors* começa a partir do minuto 40

<sup>46</sup>ele clama que isso é provavelmente o maior efeito encontrado na história das ciências sociais

<sup>47</sup> $\log(\text{chance}) = \frac{e^{0.6}}{1+e^{0.6}} \approx 3.644$

## *Priori Informativa (Informative)*

Em alguns cenários é interessante usarmos uma *priori* informativa. Geralmente são cenários que os dados são raros ou custosos de serem obtidos e que temos já algum conhecimento prévio sobre o fenômeno:

- $\text{Normal}(5, 20)$
- $\text{Log-Normal}(0, 5)$
- $\text{Beta}(100, 9803)$ <sup>48</sup>

---

<sup>48</sup>esta é usada nos modelos de COVID do grupo de pesquisa CoDatMo Stan  
Estatística Bayesiana | Jose Storopoli josees@uni9.pro.br  
Página 142

## *Prioris do rstanarm e brms*

Tanto rstanarm quanto brms possuem *prioris* padrões incorporadas nos seus modelos.

**Recomendo fortemente que você use uma *priori* específica e não se atenha às *prioris* padrões do rstanarm ou brms.**

- Elas refletem o estado-da-arte e as melhores práticas de especificação de *prioris*
- Podem mudar conforme versões dos pacotes
- Não faz o seu modelo/código ser **replicável**

# Prioris do rstanarm

Argumento	Usado em	Aplica-se à
prior_intercept	Todas funções de modelagem exceto stan_polr e stan_nlmer	Constante ( <i>intercept</i> ) do modelo, após centralização dos preditores
prior	Todas funções de modelagem	Coeficientes de Regressão, não inclui coeficientes que variam por grupo em modelos multiníveis (veja prior_covariance)
prior_aux	stan_glm, stan_glmer, stan_gamm4, stan_nlmer	Parâmetro auxiliar (ex: desvio padrão (standard error – DP), interpretação depende do modelo)
prior_covariance	stan_glmer, stan_gamm4, stan_nlmer	Matrizes de covariância em modelos multiníveis

## rstanarm – *Prioris Uniformes*

Especifica-se colocando o valor NULL (nulo em R) nos argumentos `prior_*` dos modelos `rstanarm`:

- `prior_intercept = NULL` – **constante** possuirá *priori* uniforme sobre todos os números reais  $(-\infty, +\infty)$
- `prior = NULL` – **parâmetros** possuirão *prioris* uniformes sobre todo os números reais  $(-\infty, +\infty)$
- `prior_aux = NULL` – **parâmetros auxiliares** (geralmente o erro do modelo) possuirão *prioris* uniforme sobre todos os números reais  $(-\infty, +\infty)$ <sup>49</sup>.

---

<sup>49</sup>No caso de erro do modelo, isto se restringe aos numeros reais positivos:  $[0, +\infty)$

# *rstanarm – Priors Uniformes*

```
stan_glm(y ~ ...,
          prior = NULL,
          prior_intercept = NULL,
          prior_aux = NULL)
```

## rstanarm – Prioris Informativas

Coloca-se qualquer distribuição nos argumentos `prior_*` dos modelos rstanarm:

- `prior = normal(0, 5)` – `Normal(0, 5)`
- `prior_intercept = student_t(4, 0, 10)` – `Student(\nu = 4, 0, 10)`
- `prior_aux = cauchy(0, 3)` – `Cauchy+(0, 3)`<sup>50</sup>

---

<sup>50</sup>Note que como ela é especificada como uma *priori* de um parâmetro auxiliar (no caso o erro do modelo), ela somente tomará valores positivos por isso o Cauchy<sup>+</sup>

# rstanarm – *Prioris Informativas*

```
stan_glm(y ~ ... ,  
         prior = normal(0, 5) ,  
         prior_intercept = student_t(4, 0, 10) ,  
         prior_aux = cauchy(0, 3)  
     )
```

# rstanarm – Prioris Padrões em Modelos Gaussianos

- **Constante (Intercept):** Normal centralizada com média  $\mu_y$  e desvio padrão de  $2.5\sigma_y$  - prior\_intercept = normal(**mean\_y**, 2.5 \***sd\_y**)
- **Coeficientes:** Normal para cada coeficiente média  $\mu = 0$  e desvio padrão de  $2.5 \times \frac{\sigma_y}{\sigma_x}$  - prior = normal(0, 2.5 \* **sd\_y/sd\_xk**)

Em notação matemática:

$$\begin{aligned}\alpha &\sim \text{Normal}(\mu_y, 2.5 \cdot \sigma_y) \\ \beta &\sim \text{Normal} \left( 0, 2.5 \cdot \frac{\sigma_y}{\sigma_x} \right) \\ \sigma &\sim \text{Exponential} \left( \frac{1}{\sigma_y} \right)\end{aligned}$$

# rstanarm – Prioris Padrões em Modelos Lineares Generalizados

- **Constante (Intercept):** Normal centralizada com média  $\mu = 0$  e desvio padrão de  $2.5\sigma_y$  - prior\_intercept = normal(0, 2.5 \* **sd\_y**)
- **Coeficientes:** Normal para cada coeficiente média  $\mu = 0$  e desvio padrão de  $2.5 \times \frac{1}{\sigma_x}$  - prior = normal(0, 2.5 \* **1/sd\_xk**)

Em notação matemática:

$$\alpha \sim \text{Normal}(0, 2.5 \cdot \sigma_y)$$

$$\beta \sim \text{Normal} \left( 0, 2.5 \cdot \frac{1}{\sigma_x} \right)$$

$$\sigma \sim \text{Exponential} \left( \frac{1}{\sigma_y} \right)$$

## Prioris do brms

rstanarm faz com que as *prioris* sejam automaticamente transformadas para possuírem média 0 e desvio padrão/variância 1. O resultado disso é que a **escala das prioris são unitárias**.

brms **não faz isso**. Por padrão, as *prioris* não são transformadas, portanto, **se atente com a escala as prioris do brms**.

# brms – Prioris Informativas

```
brm(y ~ ... ,  
prior = c(  
  set_prior("normal(0, 5)", class = "b", coef = "...") ,  
  ...  
  set_prior("student_t(4, 0, 10)", class = "b", coef = "intercept") ,  
  set_prior("cauchy(0, 3)", class = "sigma")  
 )  
)
```

## brms – *Prioris* Padrões

Como o brms dá muito mais autonomia, poder e flexibilidade ao usuário, todas as *prioris* padrões dos coeficientes dos modelos são literalmente *prioris* uniformes sobre todo os números reais  $(-\infty, +\infty)$ . brms apenas ajusta de maneira robusta a *priori* para:

- Constante (*Intercept*):  $t$  de Student média  $\mu = \text{mediana}(y)$ , desvio padrão de  $\text{MAD}(y)$  e graus de liberdade  $\nu = 3$
- Erro (*sigma*):  $t$  de Student média  $\mu = 0$  e desvio padrão de  $\text{MAD}(y)$ <sup>51</sup>

Em notação matemática:

$$\alpha \sim \text{Student}(\nu = 3, \text{mediana}(y), \text{MAD}(y))$$

$$\beta \sim \text{Unif}(-\infty, +\infty)$$

$$\sigma \sim \text{Student}(\nu = 3, 0, \text{MAD}(y))$$

---

<sup>51</sup>Median Absolute Deviation – Desvio Absoluto Mediano

# Sumário para Verificações Preditivas

5.1 Leituras Recomendadas

5.2 All models are wrong

5.3 Fluxo de Trabalho Bayesiano (*Workflow*)<sup>52</sup>

5.4 Verificação Preditiva da *Priori* (*Prior Predictive Check*)

5.4.1 Verificação Preditiva da *Priori* no rstanarm

5.4.2 Verificação Preditiva da *Priori* no brms

5.5 Verificação Preditiva da Posterior (*Posterior Predictive Check*)

5.5.1 Verificação Preditiva da Posterior no rstanarm

5.5.2 Verificação Preditiva da Posterior no brms

---

<sup>52</sup>baseado em Gelman, Vehtari et al. (2020)

# Verificações Preditivas - Leituras Recomendadas

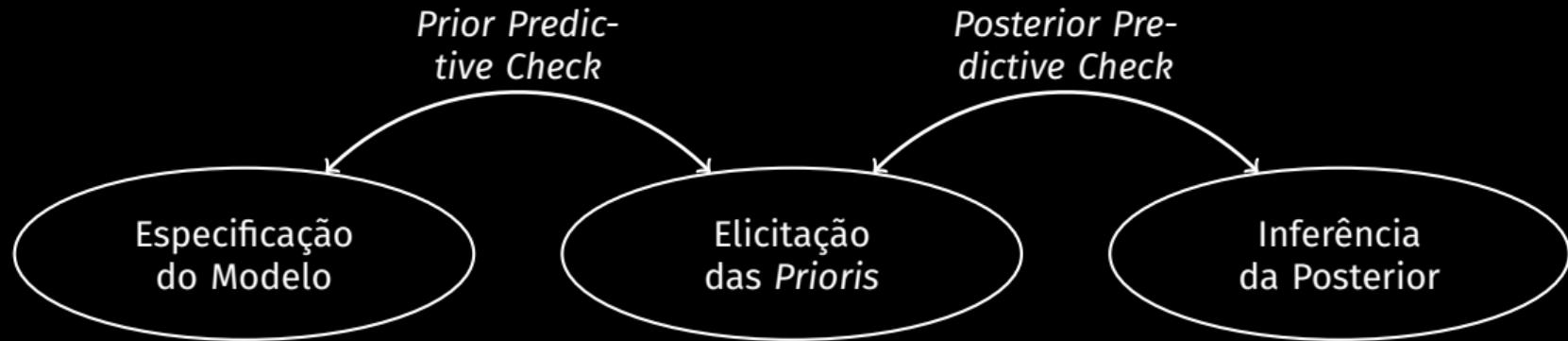
- Gelman et al. (2013b) - Capítulo 6: Model checking
- McElreath (2020) - Capítulo 5: Geocentric Models
- Gelman, Hill e Vehtari (2020):
  - Capítulo 6: Background on regression modeling
  - Capítulo 11: Assumptions, diagnostics, and model evaluation
- Gelman, Vehtari et al. (2020)

# *All models are wrong*

*All models are wrong but some are useful*  
Box (1976)



# Fluxo de Trabalho Bayesiano (*Workflow*)<sup>53</sup>



<sup>53</sup>baseado em Gelman, Vehtari et al. (2020)

## Verificação Preditiva da *Priori* (*Prior Predictive Check*)

Em especial, antes de começar a alimentar o modelo com dados precisamos fazer uma checagem de todas as nossas *prioris*.

De maneira muito simples, consiste em simular parâmetros com base nas suas distribuições especificadas *a priori* no modelo sem qualquer condicionamento aos dados e sem envolvimento nenhum da função de verossimilhança.

Independentemente do nível de informação especificada na *priori*, é sempre importante realizar uma análise de sensibilidade prévia para entender completamente a influência que as *prioris* têm na posterior.

# Verificação Preditiva da *Priori* no rstanarm e brms

- rstanarm: em qualquer função `rstan_*`() usar o argumento `prior_PD = TRUE`
- brms: na função `brm()` usar o argumento `sample_prior = "only"`

# Verificação Preditiva da *Priori* no rstanarm

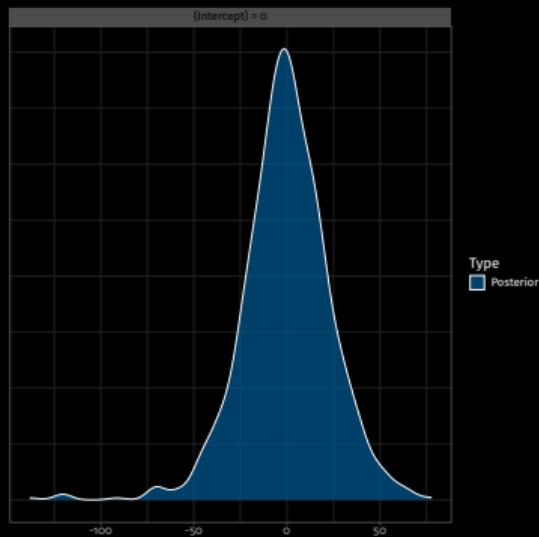
```
stan_glm(y ~ ... ,  
         prior = normal(c(0, 0), c(5, 6)),  
         prior_intercept = student_t(4, 0, 10),  
         prior_aux = cauchy(0, 3),  
         prior_PD = TRUE)
```

# Verificação Preditiva da *Priori* no brms

```
brm(y ~ x1 + x2,  
prior = c(  
  prior(normal(0, 5), class = b, coef = x1),  
  prior(normal(0, 6), class = b, coef = x2),  
  prior(student_t(4, 0, 10), class = Intercept),  
  prior(cauchy(0, 3), class = sigma)  
),  
sample_prior = "only")
```

# Verificação Preditiva da *Priori* no brms

O interessante do brms é que conseguimos naturalmente visualizar hipóteses sobre os valores do parâmetros de modelos estimados pela função brm()



# Verificação Preditiva da Posterior (*Posterior Predictive Check*)

Precisamos nos certificar que a nossa distribuição posterior de  $y$  consegue capturar todas as nuances da densidade real de  $y$ .

Isto é um procedimento chamado de Verificação Preditiva da Posterior (*Posterior Predictive Check*) e é geralmente auferido com uma inspeção visual<sup>54</sup> da densidade real de  $y$  contrastada com amostragens da densidade posterior de  $y$  estimada pelo modelo Bayesiano.

O propósito é comparar o histograma da variável dependente  $y$  contra o histograma variáveis dependentes simuladas pelo modelo  $y_{rep}$  após a estimação dos parâmetros. A ideia é que os histogramas reais e simulados se misturem e não haja divergências.

---

<sup>54</sup>também fazemos inspeções matemáticas probabilísticas, veja a seção de Comparação de Modelos Estatística Bayesiana | Jose Storopoli josees@uni9.pro.br  
Página 163

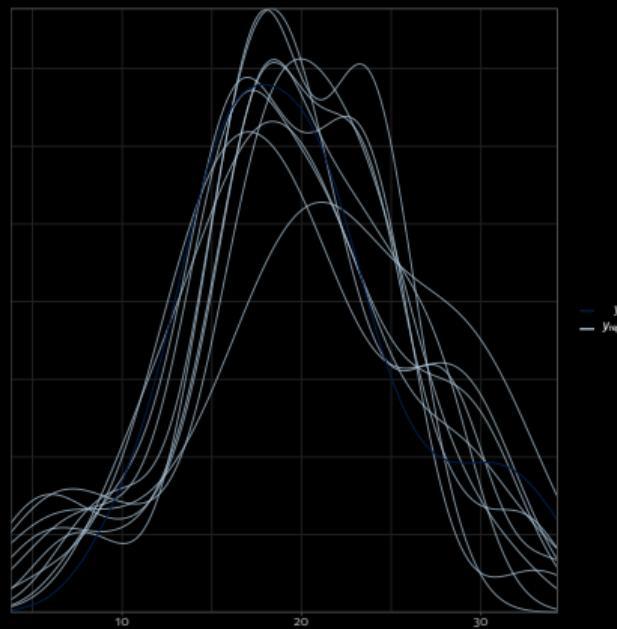
# Verificação Preditiva da Posterior no rstanarm e brms

- rstanarm: função `pp_check()` em qualquer modelo oriundo das funções `stan_*`()
- brms: função `pp_check()` em qualquer modelo oriundo da função `brm()`

# Verificação Preditiva da Posterior no rstanarm

```
rstanarm_fit <- stan_glm(mpg ~ wt + am, data = mtcars)  
pp_check(rstanarm_fit)
```

# Verificação Preditiva da Posterior no rstanarm

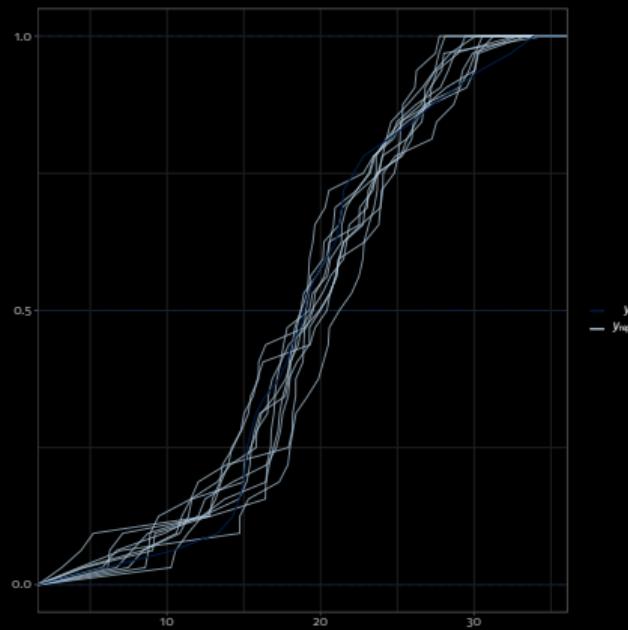
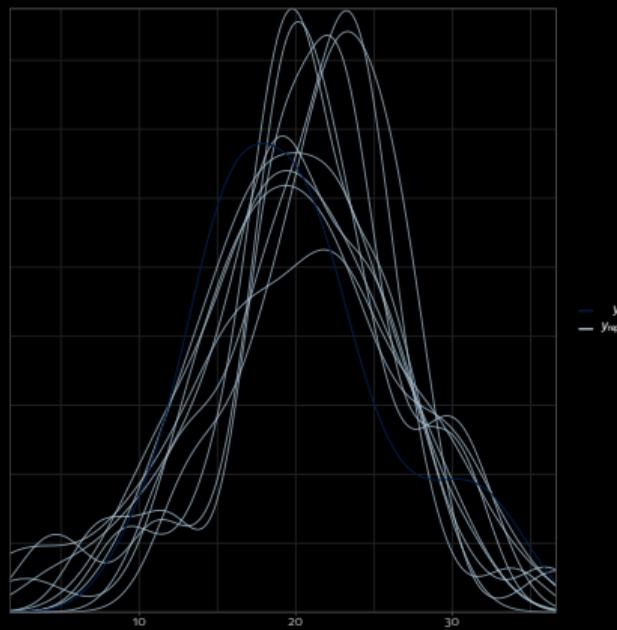


# Verificação Preditiva da Posterior no brms

O interessante do brms é que conseguimos olhar o PPC da distribuição CDF empírica (*ECDF*) também:

```
brms_fit <- brm(mpg ~ wt + am, data = mtcars)
pp_check(brmsfit)
pp_check(brmsfit, type = "ecdf_overlay")
```

# Verificação Preditiva da Posterior no brms



# Sumário para Regressão Linear

- 6.1 Leituras Recomendadas
- 6.2 Especificação da Regressão Linear
- 6.3 Regressão Linear no `rstarnarm`
- 6.4 Regressão Linear no `brms`

# Regressão Linear - Leituras Recomendadas

- Gelman et al. (2013b):
  - Capítulo 14: Introduction to regression models
  - Capítulo 16: Generalized linear models
- McElreath (2020) - Capítulo 4: Geocentric Models
- Gelman, Hill e Vehtari (2020):
  - Capítulo 7: Linear regression with a single predictor
  - Capítulo 8: Fitting regression models
  - Capítulo 10: Linear regression with multiple predictors
- Storopoli (2021) - Regressão Linear
- Tutorial de `rstanarm` de Muth et al. (2018)
- Vinheta do `rstanarm` sobre Modelos Lineares Contínuos

# O que é Regressão Linear?

Vamos falar de um classe de modelo conhecido como regressão linear. A ideia aqui é modelar uma variável dependente sendo a combinação linear de variáveis independentes.

$$\mathbf{y} = \alpha + \mathbf{X}\beta + \epsilon$$

Sendo que:

- $y$  – variável dependente
- $\alpha$  – constante (também chamada de *intercept*)
- $\beta$  – vetor de coeficientes
- $X$  – matriz de dados
- $\epsilon$  – erro do modelo

# Especificação da Regressão Linear

Para estimar a constante  $\alpha$  e os coeficientes  $\beta$  usamos uma função de verosimilhança Gaussiana/normal. Matematicamente o modelo de regressão Bayesiano é:

$$\mathbf{y} \sim \text{Normal}(\alpha + \mathbf{X}\boldsymbol{\beta}, \sigma)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\boldsymbol{\beta} \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

$$\sigma \sim \text{Exponencial}(\lambda_\sigma)$$

# Especificação da Regressão Linear

O que falta é especificar quais são as *prioris* dos parâmetros do modelo:

- Distribuição *priori* de  $\alpha$  – Conhecimento que temos da constante do modelo
- Distribuição *priori* de  $\beta$  – Conhecimento que temos dos coeficientes das variáveis independentes do modelo
- Distribuição *priori* de  $\sigma$  – Conhecimento que temos sobre o erro do modelo.

Importante que o erro pode ser somente positivo. Além disso é intuitivo colocar uma distribuição que dê peso maior para valores próximos de zero, mas que permita também valores distantes de zero, portanto uma distribuição com cauda longa é bem-vinda. Distribuições candidatas são a Exponencial que só tem suporte nos números reais positivos (então já resolve a questão de erros negativos) ou a Cauchy<sup>+</sup> truncada para apenas números positivos<sup>55</sup>

---

<sup>55</sup> lembrando que a distribuição Cauchy é a *t* de Student com graus de liberdade  $\nu = 1$

# Especificação da Regressão Linear

O nosso objetivo é **encontrar a distribuição posterior dos parâmetros de interesse** do modelo ( $\alpha$  e  $\beta$ ) calculando a distribuição posterior completa de:

$$P(\boldsymbol{\theta} \mid \mathbf{y}) = P(\alpha, \beta, \sigma \mid \mathbf{y})$$

# Regressão Linear no rstanarm

Usamos a função `stan_glm()` com o argumento `family = gaussian(link = "identity")`:

```
modelo_linear <- stan_glm(  
  y ~ ... ,  
  data = df,  
  family = gaussian(link = "identity") ,  
  prior = ... ,  
  prior_intercept = ... ,  
  prior_aux = ...  
)
```

# Regressão Linear no brms

Usamos a função `brm()` com o argumento `family = gaussian(link = "identity")`:

```
modelo_linear <- brm(  
  y ~ ... ,  
  data = df,  
  family = gaussian(link = "identity") ,  
  prior = c(  
    set_prior(..., class = "b", coef = "..."),  
    ...  
    set_prior(..., class = "b", coef = "intercept"),  
    set_prior(..., class = "sigma")  
  )  
)
```

# Sumário para Regressão Logística

7.1 Leituras Recomendadas

7.2 Dados Binários

7.3 O que é Regressão Logística?

7.3.1 Função Logit

7.3.2 Função Probit

7.3.3 Função Logística versus Função Probit

7.4 Comparativo com a Regressão Linear

7.5 Especificação da Regressão Logística

7.5.1 Verossimilhança Bernoulli

7.5.2 Verossimilhança Binomial

7.6 Intepretação dos Coeficientes

7.7 Regressão Logística no `rstarnarm`

7.8 Regressão Logística no `brms`

# Regressão Logística - Leituras Recomendadas

- Gelman et al. (2013b) - Capítulo 16: Generalized linear models
- McElreath (2020):
  - Capítulo 10: Big Entropy and the Generalized Linear Model
  - Capítulo 11, Seção 11.1: Binomial regression
- Gelman, Hill e Vehtari (2020):
  - Capítulo 13: Logistic regression
  - Capítulo 14: Working with logistic regression
  - Capítulo 15, Seção 15.3: Logistic-binomial model
  - Capítulo 15, Seção 15.4: Probit regression
- Storopoli (2021) - Regressão Logística
- Tutorial de rstanarm de Muth et al. (2018)
- Vinheta do rstanarm sobre Modelos Lineares Generalizados com dados Binários

# Bem-Vindo ao Mundo Mágico dos Modelos Lineares Generalizados

Saindo do universo dos modelos lineares, começamos a nos aventurar nos modelos lineares generalizados (*generalized linear models* – GLM).

O primeiro deles é a **regressão logística** (também chamada de regressão binomial).

# Dados Binários<sup>56</sup>

Usamos regressão logística quando a nossa variável dependente é **binária**. Ela possui apenas dois valores distintos, geralmente codificados como 0 ou 1.

---

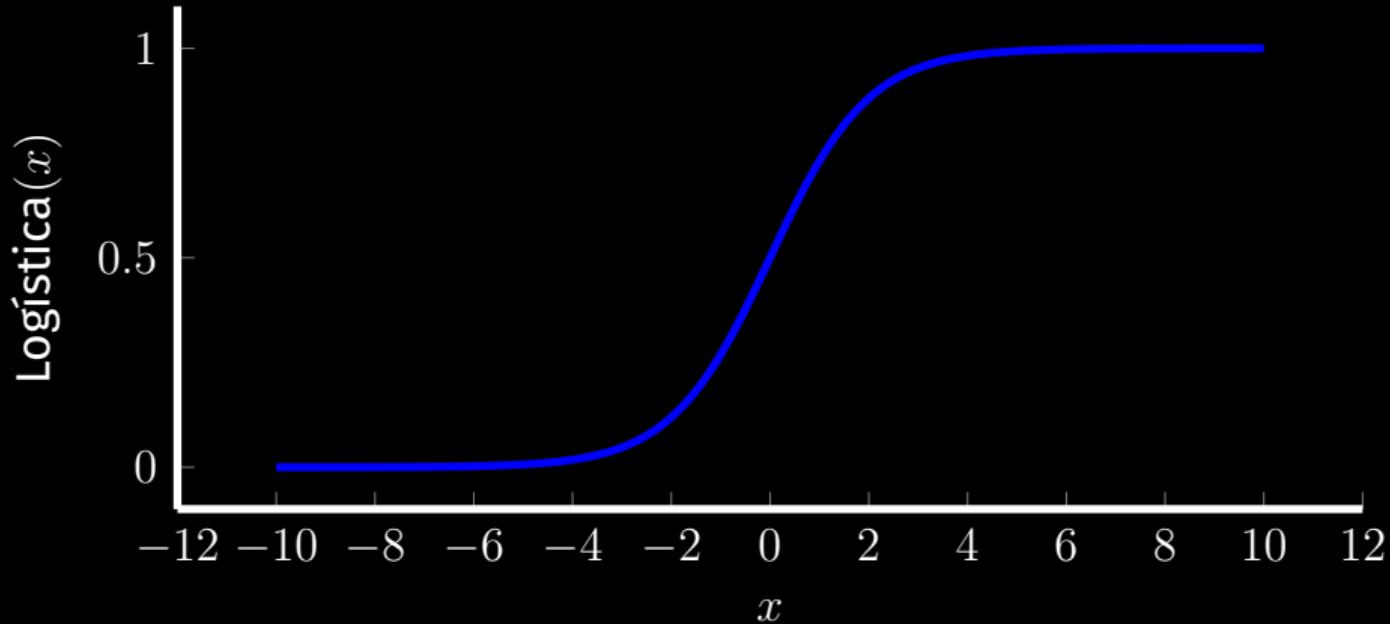
<sup>56</sup>também conhecido como dicotômico, *dummy*, etc.

# O que é Regressão Logística?

Uma regressão logística se comporta exatamente como um modelo linear: faz uma predição simplesmente computando uma soma ponderada das variáveis independentes  $X$  pelos coeficientes estimados  $\beta$ , mais uma constante  $\alpha$ . Porém ao invés de retornar um valor contínuo  $y$ , como a regressão linear, retorna a **função logística** desse valor:

$$\text{Logística}(x) = \frac{1}{1 + e^{-x}}$$

# Função Logística

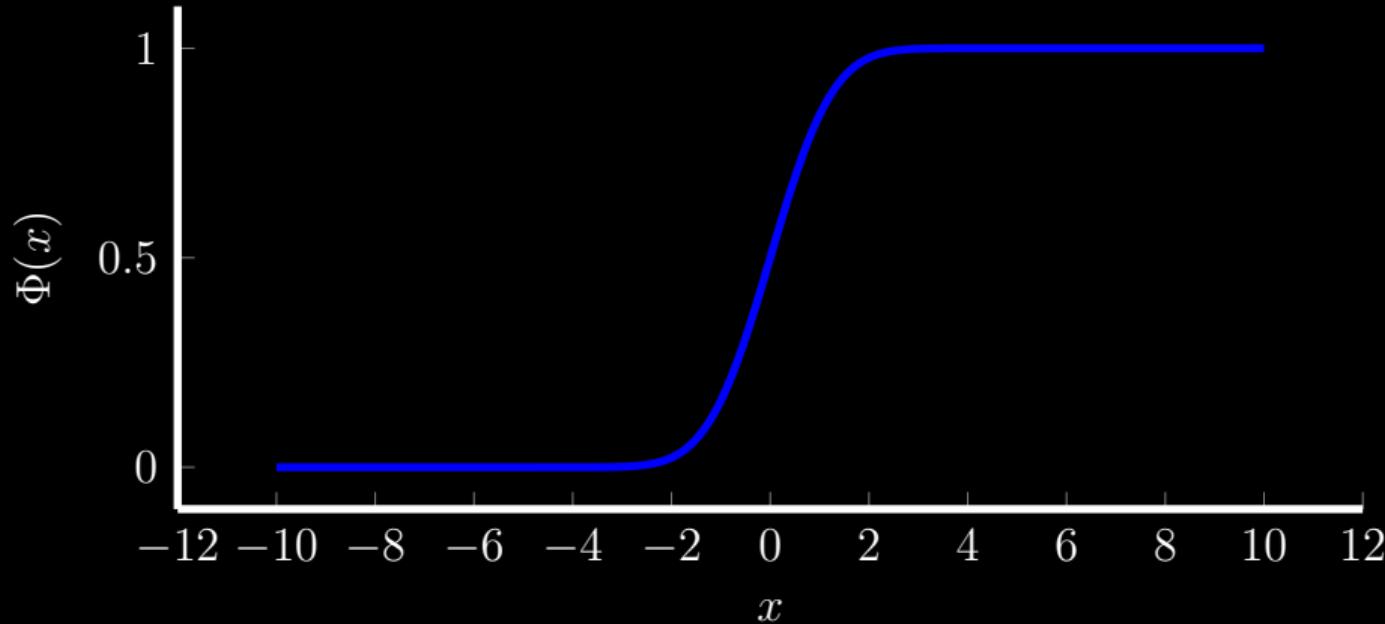


# Função Probit

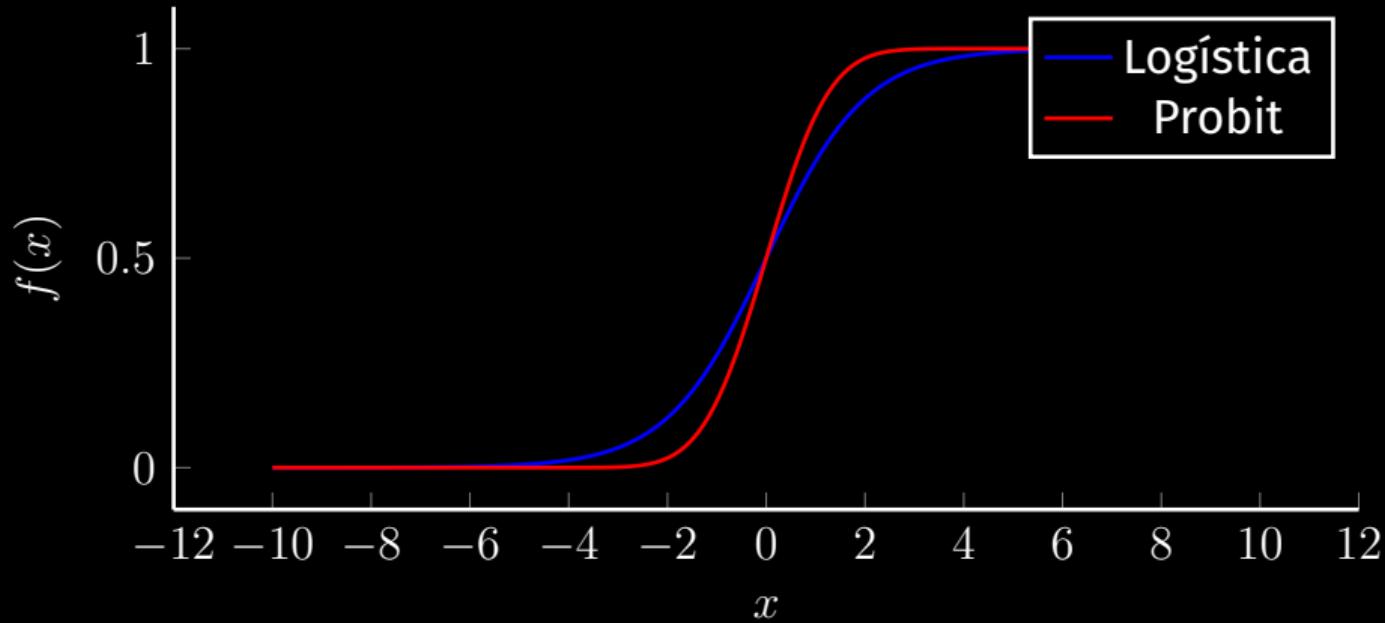
Às vezes podemos também usar a **função probit** (usualmente representada pela letra grega  $\Phi$ ) que é a CDF da distribuição Normal:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$$

# Função Probit



# Função Logística versus Função Probit



# Comparativo com a Regressão Linear

A regressão linear segue a seguinte formulação matemática:

$$\text{Linear} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Onde:

- $\alpha$  - constante
- $\beta = \beta_1, \beta_2, \dots, \beta_k$  - coeficientes das variáveis independentes  $x_1, x_2, \dots, x_k$
- $k$  - número de variáveis independentes

Se você implementar uma pequena gambiarra matemática, você terá a **regressão logística**:

- $\hat{p} = \text{Logística}(\text{Linear}) = \frac{1}{1+e^{-\text{Linear}}}$  - probabilidade prevista da observação ser o valor 1
- $\hat{y} = \begin{cases} 0 & \text{se } \hat{p} < 0.5 \\ 1 & \text{se } \hat{p} \geq 0.5 \end{cases}$  - previsão do valor discreto de  $y$

# Especificação da Regressão Logística

Podemos modelar regressão logística de duas maneiras:

- com a **verossimilhança Bernoulli** modelamos uma variável dependente **binária**  $y$  que é o resultado de um experimento de Bernoulli com uma certa probabilidade  $p$ .
- com a **verossimilhança binomial** modelamos uma variável dependente **contínua**  $y$  que é o número de sucessos de  $n$  experimentos Bernoulli independentes.

# Verossimilhança Bernoulli

$y \sim \text{Bernoulli}(p)$

$p \sim \text{Logística}/\text{Logit}(\alpha + \mathbf{X}\beta)$

$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$

$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$

Sendo que:

- $y$  - **variável dependente binária**
- $p$  - probabilidade de  $y$  tomar o valor de 1 - sucesso de um experimento Bernoulli independente
- Logística/Logit - função logística ou logit
- $\alpha$  - constante (também chamada de *intercept*)
- $\beta$  - vetor de coeficientes
- $\mathbf{X}$  - matriz de dados

# Verossimilhança Binomial

$$y \sim \text{Binomial}(n, p)$$

$$p \sim \text{Logística/Probit}(\alpha + \mathbf{X}\beta)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

Sendo que:

- $y$  - **variável dependente contínua** - sucessos de  $n$  experimentos Bernoulli independentes
- $n$  - número de experimentos Bernoulli independentes
- $p$  - probabilidade de  $y$  tomar o valor de  $y$  - sucesso de um experimento Bernoulli independente
- Logística/Logit - função logística ou logit
- $\alpha$  - constante (também chamada de *intercept*)
- $\beta$  - vetor de coeficientes
- $\mathbf{X}$  - matriz de dados

# Especificação da Regressão Logística

O nosso objetivo é **encontrar a distribuição posterior dos parâmetros de interesse** do modelo ( $\alpha$  e  $\beta$ ) calculando a distribuição posterior completa de:

$$P(\boldsymbol{\theta} \mid \mathbf{y}) = P(\alpha, \beta \mid \mathbf{y})$$

# Interpretação dos Coeficientes

Ao vermos a fórmula de regressão logística percebemos a interpretação dos coeficientes requer uma transformação. A transformação que precisamos fazer é a que inverte a função logística.

# Probabilidade versus Chances<sup>57</sup>

Mas antes preciso falar sobre **qual a diferença matemática entre probabilidade e chances.**

- **Probabilidade:** um número real entre 0 e 1 que representa a certeza de que um evento irá acontecer por meio de frequências de longo-prazo (probabilidade frequentista) ou níveis de credibilidade (probabilidade Bayesiana).
- Chances é um número positivo real ( $\mathbb{R}^+$ ) que mensura também a certeza de um evento. Mas essa certeza não é expressa como uma probabilidade (algo entre 0 e 1), mas como uma **razão entre a quantidade de resultados que produzem o evento desejado e a quantidade de resultados que não produzem o evento desejado:**

$$\text{Chances} = \frac{p}{1 - p}$$

onde  $p$  é a probabilidade.

---

<sup>57</sup>em inglês *probability* e *odds*

# Probabilidade versus Chances

$$\text{Chances} = \frac{p}{1-p}$$

onde  $p$  é a probabilidade.

- Chance com o valor de 1 é uma chance neutra algo como uma moeda justa  $p = \frac{1}{2}$
- Chances abaixo de 1 decrescem a probabilidade de vermos um certo evento
- Chances acima de 1 aumentam a probabilidade do evento.

# Log das Chances<sup>58</sup>

Se você revisitar a função logística, verá que ela tanto a constante quanto os coeficientes de  $\beta$  são literalmente o log da chance:

$$p \sim \text{Logística/Logit}(\alpha + \mathbf{X}\beta)$$

$$p \sim \text{Logística/Logit}(\alpha) + \text{Logística/Logit}(\mathbf{X}\beta)$$

$$\beta = \frac{1}{1 + e^{(-\beta)}}$$

$$\beta = \log(\text{Chance})$$

---

<sup>58</sup>em inglês *logodds*

# Log das Chances

Portanto, os coeficientes de uma regressão logística são expressados em *logodds* no qual 0 é o elemento neutro e qualquer número acima ou abaixo aumenta ou diminui as chances de obtermos um "sucesso" de  $y$ . Para termos uma interpretação mais intuitiva (igual a das casas de apostas) precisamos converter as *logodds* em chances revertendo a função log. Para isso basta "exponenciar" os valores de  $\alpha$  e  $\beta$ :

$$\text{Chances}(\alpha) = e^\alpha$$

$$\text{Chances}(\beta) = e^\beta$$

# Regressão Logística no rstanarm

Usamos a função `stan_glm()` com os argumentos `family = binomial(link = "logit")` ou `family = binomial(link = "probit")`:

```
modelo_binomial <- stan_glm(  
  y ~ ... ,  
  data = df,  
  family = binomial(link = "logit"), # ou link = "probit"  
  prior = ... ,  
  prior_intercept = ...  
)
```

# Regressão Logística no brms

Usamos a função `brm()` com os argumentos `family = binomial(link = "logit")` ou `family = binomial(link = "probit")`:

```
modelo_binomial <- brm(  
  y ~ ... ,  
  data = df,  
  family = binomial(link = "logit"), # ou link = "probit"  
  prior = c(  
    set_prior(..., class = "b", coef = "..."),  
    ...  
    set_prior(..., class = "b", coef = "intercept")  
  )  
)
```

# Sumário para Regressão de Poisson

8.1 Leituras Recomendadas

8.2 Dados de Contagem<sup>59</sup>

8.3 O que é Regressão de Poisson?

8.3.1 Função Exponencial

8.4 Comparativo com a Regressão Linear

8.5 Especificação da Regressão de Poisson

8.6 Intepretação dos Coeficientes

8.7 Regressão de Poisson no `rstarnarm`

8.8 Regressão de Poisson no `brms`

---

<sup>59</sup>*count data*

# Regressão de Poisson - Leituras Recomendadas

- Gelman et al. (2013b) - Capítulo 16: Generalized linear models
- McElreath (2020):
  - Capítulo 10: Big Entropy and the Generalized Linear Model
  - Capítulo 11, Seção 11.2: Poisson regression
- Gelman, Hill e Vehtari (2020) - Capítulo 15, Seção 15.2: Poisson and negative binomial regression
- Storopoli (2021) - Regressão de Poisson
- Tutorial de `rstanarm` de Muth et al. (2018)
- Vinheta do `rstanarm` sobre Modelos Lineares Generalizados com dados de Contagem

# Bem-Vindo ao Mundo Mágico dos Modelos Lineares Generalizados

Saindo do universo dos modelos lineares, começamos a nos aventurar nos modelos lineares generalizados (*generalized linear models – GLM*).

O segundo deles é a **regressão de Poisson**.

# Dados de Contagem

Regressão de Poisson é usada quando a nossa variável dependente só pode tomar **valores positivos**, geralmente em contextos de **dados de contagem**.

# O que é Regressão de Poisson?

Uma regressão de Poisson se comporta exatamente como um modelo linear: faz uma predição simplesmente computando uma soma ponderada das variáveis independentes  $X$  pelos coeficientes estimados  $\beta$ ,  $y$ , como a regressão linear, retorna o **logaritmo natural** desse valor:

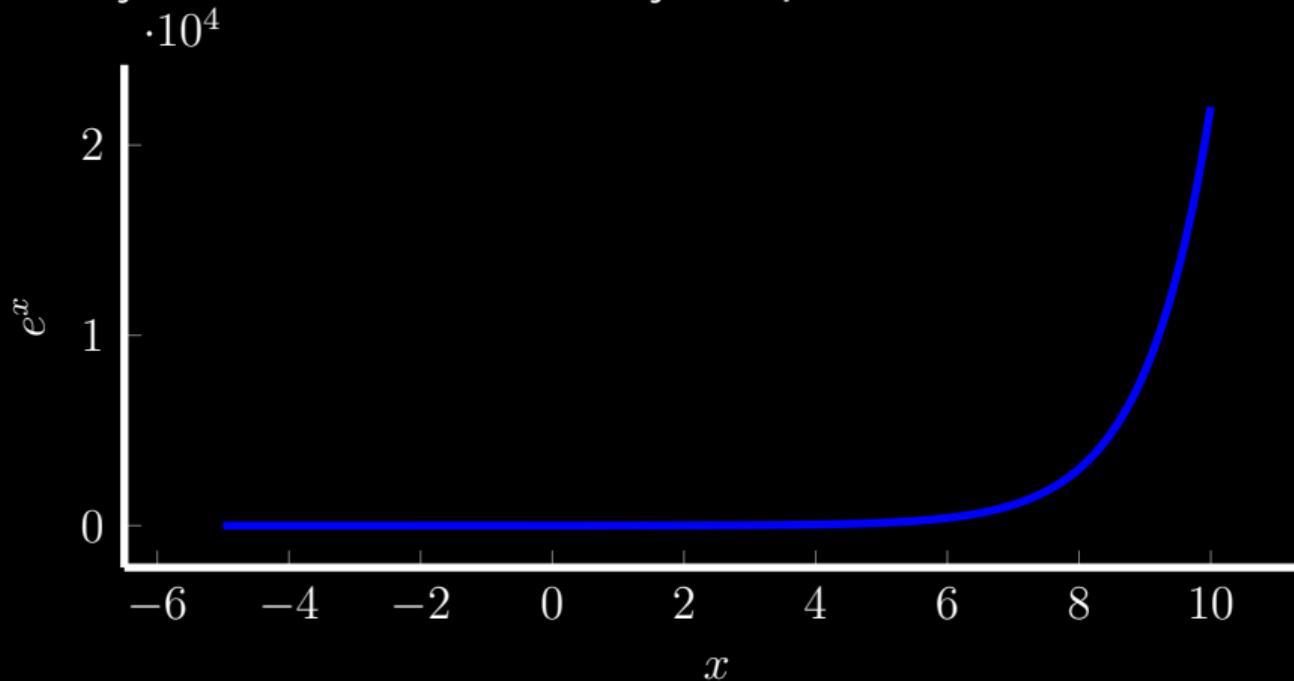
$$\log(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

que é o mesmo que:

$$y = e^{(\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)}$$

# Função Exponencial

A função  $e^x$  é chamada de função exponencial:



# Comparativo com a Regressão Linear

A regressão linear segue a seguinte formulação matemática:

$$\text{Linear} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

Onde:

- $\alpha$  - constante
- $\beta = \beta_1, \beta_2, \dots, \beta_k$  - coeficientes das variáveis independentes  $x_1, x_2, \dots, x_k$
- $k$  - número de variáveis independentes

Se você implementar uma pequena gambiarra matemática, você terá a **regressão de Poisson**:

- $\log y = e^{\text{Linear}} = e^{\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k}$

# Especificação da Regressão de Poisson

Podemos fazer uma regressão de Poisson se a variável dependente  $y$  for uma variável com dados de contagem, ou seja,  $y$  somente toma valores positivos. A função de **verossimilhança de Poisson** usa uma constante  $\alpha$  e os coeficientes  $\beta$  porém estes são "exponenciados" ( $e^x$ ):

$$y \sim \text{Poisson}(e^{(\alpha + \mathbf{X}\beta)})$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

# Interpretação dos Coeficientes

Ao vermos a fórmula de regressão de Poisson percebemos a interpretação dos coeficientes requer uma transformação. A transformação que precisamos fazer é a que inverte a função logarítmica:

$$\log^{-1}(x) = e^x$$

Então precisamos novamente "exponenciar" os valores de  $\alpha$  e  $\beta$ :

$$\begin{aligned} \mathbf{y} &= e^{(\alpha + \mathbf{X}\boldsymbol{\beta})} \\ &= e^\alpha \cdot e^{(X_{(1)} \cdot \beta_{(1)})} \cdot e^{(X_{(2)} \cdot \beta_{(2)})} \cdot \dots \cdot e^{(X_{(k)} \cdot \beta_{(k)})} \end{aligned}$$

# Regressão de Poisson no rstanarm

Usamos a função `stan_glm()` com o argumento `poisson(link = "log")`:

```
modelo_poisson <- stan_glm(  
  y ~ ... ,  
  data = df,  
  family = poisson(link = "log") ,  
  prior = ... ,  
  prior_intercept = ...  
)
```

# Regressão de Poisson no brms

Usamos a função `brm( )` com o argumento `poisson(link = "log")`:

```
modelo_poisson <- brm(  
  y ~ ... ,  
  data = df,  
  family = poisson(link = "log") ,  
  prior = c(  
    set_prior(..., class = "b", coef = "..."),  
    ...  
    set_prior(..., class = "b", coef = "intercept")  
  )  
)
```

# Sumário para Regressão Robusta

9.1 Leituras Recomendadas

9.2 Dados com *Outliers*

9.3 Superdispersão

9.4 Versões com Superdispersão dos Modelos Probabilísticos Padrões

9.4.1  $t$  de Student ao invés da Normal

9.4.2 Beta-Binomial ao invés da Binomial

9.4.3  $t$  de Student ao invés da Binomial

9.4.4 Binomial Negativa ao invés de Poisson

9.4.5 Mistura de Binomial Negativa ao invés de Poisson

9.5 Por quê usar Modelos Não-Robustos?

# Regressão Robusta - Leituras Recomendadas

- Gelman et al. (2013b) - Capítulo 17: Models for robust inference
- McElreath (2020) - Capítulo 12: Monsters and Mixtures
- Gelman, Hill e Vehtari (2020):
  - Capítulo 15, Seção 15.6: Robust regression using the t model
  - Capítulo 15, Seção 15.8: Going beyond generalized linear models
- Storopoli (2021) - Regressão Robusta
- Tutorial de brms de Bürkner (2018)

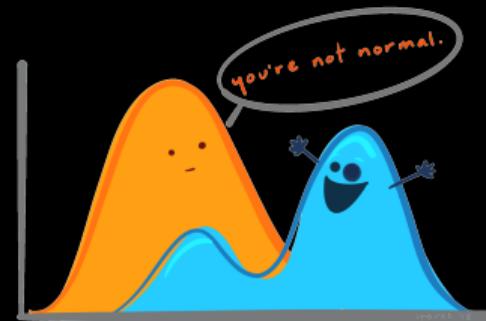
# Modelos Robustos<sup>60</sup>

Quase sempre nossos dados no mundo real são bem estranhos.

Por conveniência usamos modelos simples. Mas sempre se pergunte. De que maneiras a inferência da posterior depende de:

- Observações extremas (*outliers*)?
- Suposições de modelo não-acessíveis?

Além disso vamos usar **exclusivamente** o brms ao invés do rstanarm.



<sup>60</sup>figura de Allison Horst (CC-BY-4.0)

# Dados com *Outliers*

Modelos baseados na **distribuição normal** são notoriamente "não robustos" para outliers, no sentido de que **uma única observação outlier pode afetar fortemente a inferência para todos os parâmetros no modelo**, mesmo aqueles com pouca conexão substantiva com a observação outlier.

# Superdispersão (*Overdispersion*)

## Definição (Superdispersão e Subdispersão)

A *superdispersão overdispersion* e a *subdispersão underdispersion* referem-se a dados que mostram mais ou menos variação do que o esperado com base em um modelo de probabilidade. (Gelman, Hill & Vehtari, 2020)

Para cada um dos modelos padrão, há de fato uma **extensão natural** em que um **único** parâmetro é adicionado para permitir a superdispersão (Gelman et al., 2013b).

---

<sup>61</sup>bem mais raro no mundo real

# Exemplo de Superdispersão

## Exemplo (Acidentes de Trânsito)

*Suponha que você esteja analisando acidentes de trânsito. O modelo usualmente usado nesses tipos de fenômenos é a **Régressão de Poisson**. A distribuição de Poisson possui o mesmo valor como média e variância. Então, se você encontrar uma variação nos dados maior que a verossimilhança Poisson permite, o modelo probabilístico provavelmente não conseguirá reproduzir com fidelidade o fenômeno modelado.*

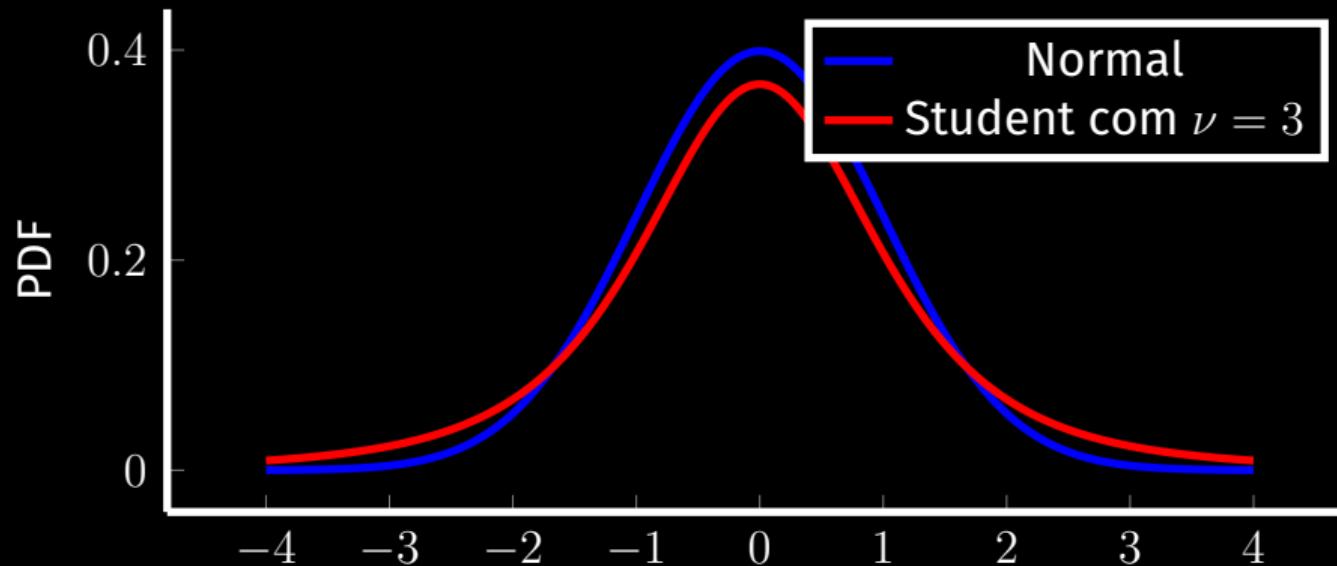
## *t* de Student ao invés da Normal

A distribuição *t* de Student tem uma **cauda mais longa** que a distribuição Normal.

O que faz uma boa candidata para **acomodar observações outliers sem gerar instabilidades na inferência dos parâmetros**.

Do ponto de vista Bayesiano, não há nada especial na verossimilhança Gaussiana/Normal. É apenas uma distribuição probabilística especificada em um modelo. Podemos deixar o modelo mais robusto ao usarmos uma distribuição *t* de Student como função de verossimilhança.

# *t* de Student ao invés da Normal



## *t* de Student ao invés da Normal

Ao usarmos uma verossimilhança *t* de Student ao invés da Normal, o erro do modelo,  $\sigma$  não segue uma distribuição normal, mas sim uma distribuição *t* de Student:

$$\mathbf{y} \sim \text{Student}(\nu, \alpha + \mathbf{X}\boldsymbol{\beta}, \sigma)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\boldsymbol{\beta} \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

$$\nu \sim \text{Log-Normal}(2, 1)$$

$$\sigma \sim \text{Exponencial}(\lambda_\sigma)$$

Além disso, é apropriado incluir os graus de liberdade  $\nu$  como um parâmetro a ser estimado pelo modelo (Gelman et al., 2013b). Uma *priori* de cauda longa e restrita a somente tomar valores positivos é adequada.

# brms – $t$ de Student ao invés da Normal

```
brm( ...  
  family = student(link = "identity")  
)
```

# Beta-Binomial ao invés da Binomial

A distribuição binomial tem uma limitação prática de que temos somente um parâmetro livre<sup>62</sup> ( $p$ ), o que implica em a **variância ser determinada pela média**. Isso faz com que a verossimilhança binomial **não seja robusta à superdispersão**.

Uma alternativa robusta é a **distribuição beta-binomial**, que, como o nome sugere, é uma **mistura beta de binomiais**. Além disso, permite com que a **variância seja diferente da média**, garantindo **robustez à superdispersão**.

---

<sup>62</sup>já que  $n$  vem dos dados

# Beta-Binomial ao invés da Binomial

A distribuição beta-binomial é uma binomial, mas a probabilidade  $p$  é parametrizada com uma distribuição Beta( $\alpha, \beta$ ). Geralmente usamos  $\alpha$  como a probabilidade  $p$  da binomial e  $\beta$  (também usado às vezes  $\phi$ ) é o parâmetro adicional para controlar superdispersão. Valores de  $\beta/\phi$  iguais a 1 fazem a beta-binomial se comportar igual a uma binomial.

$$y \sim \text{Beta-Binomial}(n, p, \phi)$$

$$p \sim \text{Logística/Probit}(\alpha + \mathbf{X}\boldsymbol{\beta})$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\boldsymbol{\beta} \sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}})$$

$$\phi \sim \text{Exponencial}(1)$$

É apropriado incluir o parâmetro de superdispersão  $\beta$  como um parâmetro a ser estimado pelo modelo (Gelman et al., 2013b; McElreath, 2020). Uma *priori* de cauda longa e restrita a somente tomar valores positivos é adequada.

# brms – Beta-Binomial ao invés da Binomial<sup>63</sup>

```
# verossimilhança customizada
beta_binomial2 <- custom_family("beta_binomial2",
  dpars = c("mu", "phi"),
  links = c("logit", "log"), lb = c(NA, 0),
  type = "int", vars = "vint1[n]")
stan_funs <- "
  real beta_binomial2_lpmf(int y, real mu, real phi, int T) {
    return beta_binomial_lpmf(y | T, mu * phi, (1 - mu) * phi);
  }
  int beta_binomial2_rng(real mu, real phi, int T) {
    return beta_binomial_rng(T, mu * phi, (1 - mu) * phi);
  }"
stanvars <- stanvar(scode = stan_funs, block = "functions")
brm(...,
  family = beta_binomial2, # verossimilhança customizada
  prior = c(..., prior(exponential(1), class = phi))
```

---

<sup>63</sup>sugiro verem essa implementação Solomon Kurz

# *t* de Student ao invés da Binomial

Também chamada de *Robit*<sup>64</sup> (Gelman et al., 2013b; Gelman, Hill & Vehtari, 2020). A ideia é "robustizar" a regressão logística com uma formulação usando dados latentes  $z$  e dar uma distribuição *t* de Student aos erros latentes  $\epsilon$ :

$$y_i = \begin{cases} 0 & \text{se } z_i < 0 \\ 1 & \text{se } z_i > 0 \end{cases}$$
$$z_i = X_i\beta + \epsilon_i$$
$$\epsilon_i \sim \text{Student}\left(\nu, 0, \sqrt{\frac{\nu - 2}{\nu}}\right)$$
$$\nu \sim \text{Gamma}(2, 0.1) \in [2, \infty)$$

O grande segredo aqui é usar uma distribuição Gamma como *priori* dos graus de liberdade  $\nu$  truncada para valor mínimo de  $\nu = 2$ . Outra opção é literalmente especificar  $\nu = 4$ .

---

<sup>64</sup>há uma bela discussão com Gelman, Vehtari e Kurz no *discourse* do Stan

# brms – t de Student ao invés da Binomial

```
stan_inv_robit <- "
real inv_robit(real y, real nu) {
  return(student_t_cdf(y, nu, 0, sqrt((nu - 2) / nu)));
}"
stanvar_inv_robit <- stanvar(scode = stan_inv_robit, block = "functions")
robit_formula <-
bf(y_c | trials(1) ~ inv_robit(eta, nu),
  nlf(eta ~ bo + b1 * x),
  bo + b1 ~ 1,
  nu ~ 1,
  nl = TRUE)
brm(formula = robit_formula,
  family = binomial("identity"),
  formula = robit_formula,
  prior = c(prior(normal(0, 1), nlpar = bo),
  prior(normal(0, 1), nlpar = b1),
  prior(gamma(2, 0.1), nlpar = nu, lb = 2)),
  stanvars = stanvar_inv_robit)
```

# Binomial Negativa ao invés de Poisson

Esse é o exemplo que falamos sobre superdispersão. A distribuição de Poisson possui o mesmo valor como média e variância.

Então, se você encontrar superdispersão, provavelmente precisará de uma alternativa robusta à Poisson. Aqui que entra a binomial negativa que "robustiza" a Poisson com um parâmetro extra  $\phi$ .

Esse parâmetro é a probabilidade de sucessos  $p$  da distribuição binomial negativa e geralmente usamos uma distribuição gamma como *priori* para que  $\phi$  cumpra a função de um parâmetro de "dispersão recíproca".

# Binomial Negativa ao invés de Poisson

$y \sim \text{Binomial Negativa}(e^{(\alpha + \mathbf{x}\beta)}, \phi)$

$\phi \sim \text{Gamma}(0.01, 0.01)$

$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$

$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$

A ideia é dar uma *priori* cauda longa para  $\phi$ , algo como  $\text{Gamma}(0.01, 0.01)$  funciona.

# brms – Binomial Negativa ao invés de Poisson

```
brm( ...  
  family = negbinomial(link = "log")  
)
```

# Mistura de Binomial Negativa ao invés de Poisson

Mesmo usando uma binomial negativa, caso a superdispersão seja muito acentuada, em especial quando temos muita **inflação de zeros** (*zero-inflated*), o seu modelo ainda pode resultar em patologias. Uma outra sugestão é usar uma mistura de binomial negativa (McElreath, 2020). Aqui,  $S_i$  é uma variável binária (*dummy*) indicando se a observação  $i$  tem valor diferente de zero ou não.  $S_i$  pode ser modelado usando uma regressão logística:

$$y \begin{cases} = 0, & \text{se } S_i = 0 \\ \sim \text{Binomial Negativa} (e^{(\alpha + \mathbf{X}\beta)}, \phi), & \text{se } S_i = 1 \end{cases}$$

$$P(S_i = 1) = \text{Logística}/\text{Logit}(\mathbf{X}\gamma)$$

$$\gamma \sim \text{Beta}(1, 1)$$

$\gamma$  é um novo conjunto de coeficientes para essa parte do modelo com *prioris* uniformes de Beta(1, 1).

# brms – Mistura de Binomial Negativa ao invés de Poisson

```
brm( ...  
  family = zero_inflated_negbinomial(link = "log") ,  
  prior = c(  
    ...  
    prior(gamma(0.01, 0.01), class = shape) ,  
    prior(beta(1, 1), class = zi)  
)
```

# Por quê usar Modelos Não-Robustos?

O **teorema do limite central** nos diz que a distribuição **normal** é uma modelo apropriado para dados que são formados como a **soma de um grande número de componentes independentes**.

Mesmo quando não estão naturalmente implícitos na estrutura de um problema, os **modelos simples não-robustos são computacionalmente convenientes**.

Claro o que deve sempre guiar a sua escolha de modelo, além da natureza específica do processo de geração de dados do fenômeno que você está estudando, é a **verificação preditiva da posterior**.

# Sumário para Modelos Multiníveis

- 10.1 Leituras Recomendadas
- 10.2 O que são Modelos Multiníveis?
- 10.3 Quando usar Modelos Multiníveis?
- 10.4 *Hiperpriori (Hyperprior)*
- 10.5 Abordagem Frequentista versus Abordagem Bayesiana
- 10.6 3 Abordagens de Modelos Multiníveis
  - 10.6.1 *Random-intercept model*
  - 10.6.2 *Random-slope model*
  - 10.6.3 *Random-intercept-slope model*
- 10.7 Modelos Multiníveis no `rstarnarm`
- 10.8 Modelos Multiníveis no `brms`

# Modelos Multiníveis - Leituras Recomendadas

- Gelman et al. (2013b):
  - Capítulo 5: Hierarchical models
  - Capítulo 15: Hierarchical linear models
- McElreath (2020):
  - Capítulo 13: Models With Memory
  - Capítulo 14: Adventures in Covariance
- Storopoli (2021) - Modelos Multiníveis
- Tutorial de `rstanarm` de Muth et al. (2018)
- Tutorial de `brms` de Bürkner (2018)
- Gelman e Hill (2007)
- Estudo de caso do Michael Betancourt sobre Modelos Hierárquicos
- Kruschke e Vanpaemel (2015)

# *I have many names...*

Modelos multiníveis também são conhecidos por vários nomes<sup>65</sup>:

- Modelos Hierárquicos (*Hierarchical Models*)
- Modelos de Efeitos Aleatórios (*Random Effects Models*)
- Modelos de Efeitos Mistos (*Mixed Effects Models*)
- Modelos de Dados em Painel (*Cross-Sectional Models*)
- Modelos de Dados Aninhados (*Nested Data Models*)

---

<sup>65</sup>para uma listagem completa veja aqui

# O que são Modelos Multiníveis?

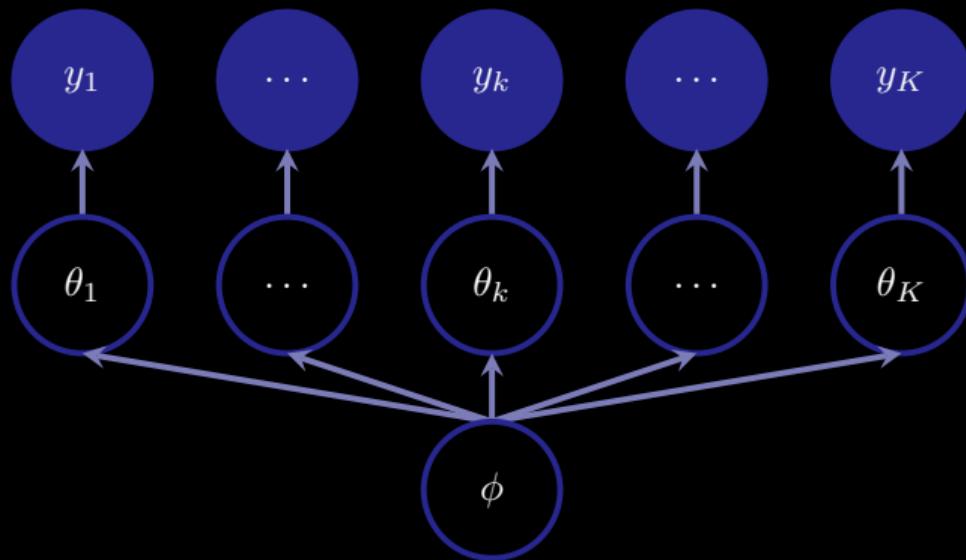
## Definição (Modelos Multiníveis)

*Modelo estatístico escrito em níveis múltiplos (forma hierárquica) que estima os parâmetros da distribuição posterior usando a abordagem Bayesiana. Os submodelos se combinam para formar o modelo hierárquico, e o teorema de Bayes é usado para integrá-los aos dados observados e contabilizar toda a incerteza que está presente.*

Os modelos hierárquicos são descrições matemáticas que envolvem vários parâmetros, de modo que as estimativas de alguns parâmetros dependem significativamente dos valores de outros parâmetros.

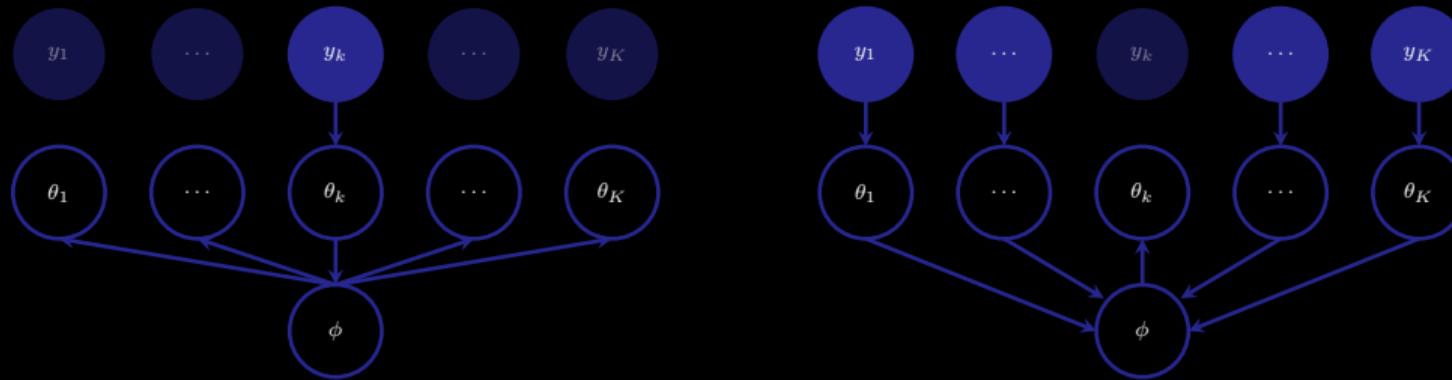
# O que são Modelos Multiníveis?

Hiperparâmetro  $\phi$  que parametriza os parâmetros  $\theta_1, \theta_2, \dots, \theta_K$  que por fim são usados para inferir a densidade posterior de alguma variável de interesse  $y = y_1, y_2, \dots, y_K$



# O que são Modelos Multiníveis?

Mesmo que as observações informem diretamente apenas um único conjunto de parâmetros, o modelo hierárquico acopla os parâmetros individuais e fornece uma porta dos fundos para que as observações informem todos os contextos.



Por exemplo, as observações do  $k$ -ésimo contexto,  $y_k$ , informam diretamente os parâmetros que quantificam o comportamento desse contexto,  $\theta_k$ . Esses parâmetros, entretanto, informam diretamente os parâmetros populacionais  $\phi$  que então informam todos os outros contextos por meio do modelo hierárquico. Da mesma forma, as observações que informam diretamente os outros contextos informam indiretamente os parâmetros populacionais que então retroalimentam o  $k$ -ésimo contexto.

# O que são Modelos Multiníveis?

A **modelagem hierárquica** é usada quando as informações estão disponíveis em vários **níveis diferentes de unidades de observação**. A forma hierárquica de análise e organização auxilia no entendimento de **problemas multiparâmetros** e também desempenha um papel importante no desenvolvimento de **estratégias computacionais**.

# Quando usar Modelos Multiníveis?

Modelos multiníveis são particularmente apropriados para projetos de pesquisa onde os dados dos participantes são organizados em mais de um nível (ou seja, dados aninhados – *nested data*). As unidades de análise geralmente são indivíduos (em um nível inferior) que estão aninhados em unidades contextuais/agregadas (em um nível superior).

Um exemplo é quando estamos mensurando desempenho de indivíduos e temos informações adicionais sobre pertencimento à grupos distintos como:

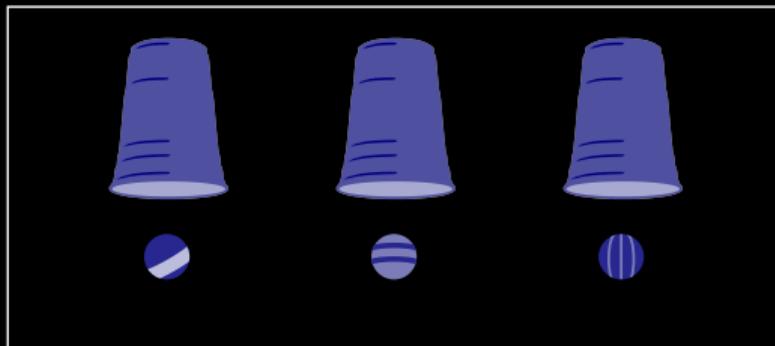
- sexo
- faixa etária
- nível hierárquico
- nível educacional
- estado/província de residência

# Quando usar Modelos Multiníveis?

O mais importante é que **não seja violado o princípio da permutabilidade** (de Finetti, 1974).

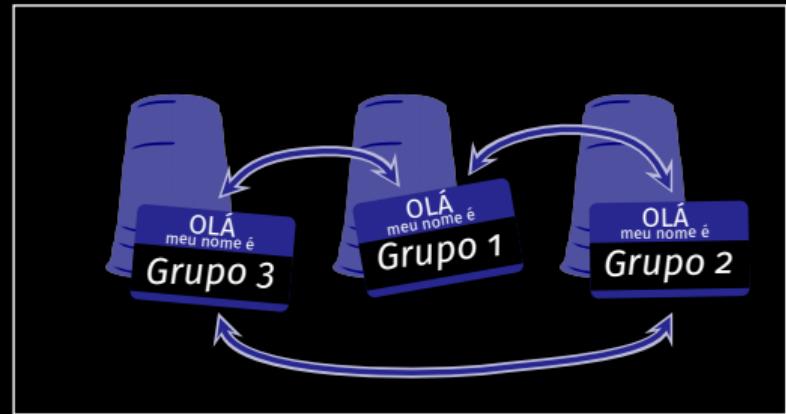
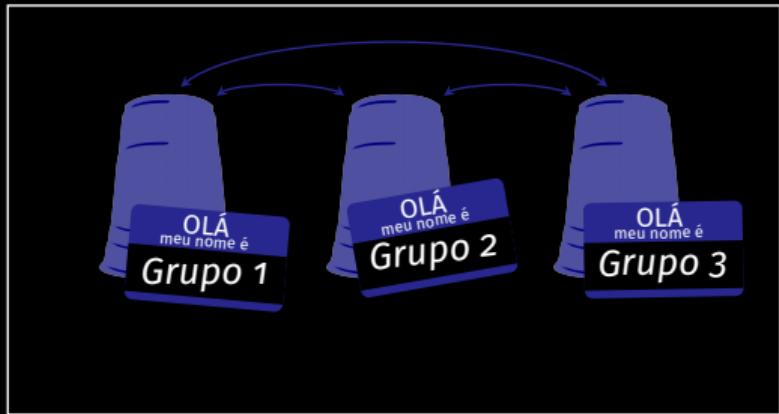
Esse pressuposto parte do princípio que os **grupos são permutáveis**.

# Revisitando a Permutabilidade (de Finetti, 1974)<sup>66</sup>



<sup>66</sup>figuras adaptadas de Michael Betancourt (CC-BY-SA-4.0)

# Revisitando a Permutabilidade (de Finetti, 1974)<sup>67</sup>



<sup>67</sup>figuras adaptadas de Michael Betancourt (CC-BY-SA-4.0)

# Hiperpriori (Hyperprior)

Em modelos multiníveis temos a figura da *hiperpriori*, que é justamente uma *priori* de uma *priori*:

$$y \sim \text{Normal}(10, \theta)$$

$$\theta \sim \text{Normal}(0, \phi)$$

$$\phi \sim \text{Exponencial}(1)$$

Aqui  $y$  são variáveis de interesse que pertencem à certos grupos distintos.  $\theta$ , uma *priori* de  $y$ , é um vetor de parâmetros de grupos com uma *priori* (que se torna *hyperpriori*)  $\phi$ .

# Abordagem Frequentista versus Abordagem Bayesiana

Existem modelos multíveis também na estatística frequentista. Todos esses estão disponíveis no pacote `lme4` (Bates et al., 2015).

- **otimização da função de verossimilhança versus aproximação da posterior via MCMC.** Quase sempre isso gera falha de convergência para modelos que não sejam extremamente simples.
- **modelos multiníveis frequentista não computam  $p$ -valores dos efeitos de grupo<sup>68</sup>.** or conta da contorção matemática de diversas aproximações que a estatística frequentista tem que fazer o cálculo de  $p$ -valores de efeitos de grupo possuem fortes pressupostos. O principal é que os grupos são balanceados. Ou seja, os grupos são homogêneos no seu tamanho. Qualquer desbalanço na composição dos grupos (um grupo com mais observações que outros) resulta em  $p$ -valores patológicos e que não podem ser confiáveis.

<sup>68</sup>veja a explicação aqui do Douglas Bates autor do pacote `lme4`

# Abordagem Frequentista versus Abordagem Bayesiana

Sumarizando, a abordagem **frequentista para modelos multiníveis não é robusta** tanto no processo da **inferência (falhas de convergência)** da estimação de máxima verossimilhança), quanto nos **resultados** dessa inferência (não computa  $p$ -valores por conta de **fortes pressupostos que quase sempre são violados**).

### 3 Abordagens de Modelos Multiníveis

- *Random-intercept model*: Modelo no qual cada grupo recebe uma constante (*intercept*) diferente além da constante global e coeficientes globais
- *Random-slope model*: Modelo no qual cada grupo recebe um coeficiente (*slope*) diferente para cada variável independente além da constante global
- *Random-intercept-slope model*: Modelo no qual cada grupo recebe tanto uma constante (*intercept*) quanto um coeficiente (*slope*) diferente para cada variável independente além da constante global

rstanarm e brms possuem as funcionalidades completas para rodar modelos multiníveis e a única coisa a se fazer é alterar a fórmula. Para rstanarm, há uma segunda mudança também que não usamos mais a função stan\_glm() mas sim a função stan\_glmer(). Para brms não há mudança e usamos a mesma função brm().

## *Random-intercept model*

A primeira abordagem é o *random-intercept model* na qual especificamos para cada grupo uma constante diferente, além da constante global. Essas constantes são amostradas de uma *hiperpriori*.

A fórmula a ser usada segue este padrão:

$$y \sim (1 \mid \text{group}) + x_1 + x_2$$

O  $(1 \mid \text{group})$  na fórmula sinaliza que a constante 1 deve ser também especificada para cada um dos grupos listados nos valores da variável group.

## *Random-intercept model*

Caso queira remover do modelo a constante global<sup>69</sup> é só especificar o 0 como constante global. Isto sinaliza que o modelo possui apenas constantes para cada grupo e que não há uma constante global a ser estimada:

$$y \sim 0 + (1 \mid \text{group}) + x_1 + x_2$$

Além disso você pode especificar uma constante para quantos grupos quiser. É só adicioná-los na fórmula:

$$y \sim (1 \mid \text{group1}) + (1 \mid \text{group2}) + x_1 + x_2$$

---

<sup>69</sup>algo que eu recomendo apenas se tiver **muita fundamentação teórica** para tal manobra

# Especificações Matemáticas dos Modelos Multiníveis

Os modelos hierárquicos geralmente são especificados assim.

Temos  $N$  observações organizadas em  $J$  grupos com  $K$  variáveis independentes.

O truque aqui é que inserimos uma coluna de 1 na matrix de dados  $\mathbf{X}$ . Matematicamente isto se comporta como se esta coluna fosse uma variável de identidade (pois o número 1 na operação de multiplicação  $1 \cdot \beta$  é o elemento identidade. Ele mapeia  $x \rightarrow x$  mantendo o valor de  $x$ ) e, consequentemente, podemos interpretar o coeficiente dessa coluna como a constante do modelo<sup>70</sup>.

---

<sup>70</sup>por isso que nas fórmulas do R o 1 é interpretado como a constante do modelo. Substitua-o por uma coluna de 0 e temos um modelo sem constante, por isso o 0 nas fórmulas é interpretado como um modelo ausente de constante

# Especificações Matemáticas dos Modelos Multiníveis

Então temos os dados como uma matriz:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1K} \\ 1 & x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{NK} \end{bmatrix}$$

# Especificação Matemática – *Random-intercept model*

Matematicamente o *Random-intercept model* para uma regressão linear é:

$$\mathbf{y} \sim \text{Normal}(\alpha + \alpha_j + \mathbf{X} \cdot \boldsymbol{\beta}, \sigma)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\alpha_j \sim \text{Normal}(0, \tau)$$

$$\boldsymbol{\beta} \sim \text{Normal}(\mu_{\boldsymbol{\beta}}, \sigma_{\boldsymbol{\beta}})$$

$$\tau \sim \text{Cauchy}^+(0, \psi_\alpha)$$

$$\sigma \sim \text{Exponential}(\lambda_\sigma)$$

## *Random-slope model*

A segunda abordagem é o *random-slope model* na qual especificamos para cada grupo um coeficiente diferente para cada variável independente desejada, além da constante global. Esses coeficientes são amostrados de uma *hiperpriori*.

A fórmula a ser usada segue este padrão:

$$y \sim (\alpha + x_1 | \text{group}) + (\alpha + x_2 | \text{group})$$

Note que usamos o  $\alpha$  pois neste caso sinalizamos que apenas a variável independente deve possuir coeficientes para cada grupo e não a constante.

# Especificação Matemática – *Random-slope model*

Matematicamente o *Random-slope model* para uma regressão linear é:

$$\mathbf{y} \sim \text{Normal}(\alpha + \mathbf{X}\boldsymbol{\beta}_j, \sigma)$$

$$\boldsymbol{\beta}_j \sim \text{Normal Multivariada}(\boldsymbol{\mu}_j, \Sigma) \quad \text{para } j \in \{1, \dots, J\}$$

$$\Sigma \sim \text{LKJ}(\eta)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\sigma \sim \text{Exponencial}(\lambda_\sigma)$$

Cada vetor de coeficientes  $\boldsymbol{\beta}_j$  representa os coeficientes das colunas de  $\mathbf{X}$  para cada grupo  $j \in J$ .

# Especificação Matemática – *Random-slope model*

Caso queira mais grupos é só adicioná-los ao modelo como  $J_1, J_2, \dots$ :

$$\mathbf{y} \sim \text{Normal}(\alpha + \mathbf{X}\boldsymbol{\beta}_{j1} + \mathbf{X}\boldsymbol{\beta}_{j2}, \sigma)$$

$$\boldsymbol{\beta}_{j1} \sim \text{Normal Multivariada}(\boldsymbol{\mu}_{j1}, \boldsymbol{\Sigma}_1) \quad \text{para } j_1 \in \{1, \dots, J_1\}$$

$$\boldsymbol{\beta}_{j2} \sim \text{Normal Multivariada}(\boldsymbol{\mu}_{j2}, \boldsymbol{\Sigma}_2) \quad \text{para } j_2 \in \{1, \dots, J_2\}$$

$$\boldsymbol{\Sigma}_1 \sim \text{LKJ}(\eta_1)$$

$$\boldsymbol{\Sigma}_2 \sim \text{LKJ}(\eta_2)$$

$$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

$$\sigma \sim \text{Exponencial}(\lambda_\sigma)$$

# Prioris para Matrizes de Covariância

Podemos especificar uma *priori* para a matriz de covariância  $\Sigma$ .

Para eficiência computacional podemos fazer a matriz de covariância  $\Sigma$  vire uma matriz de correlação. Toda matriz de covariância pode ser decomposta em:

$$\Sigma = \text{diag}_{\text{matrix}}(\tau) \cdot \Omega \cdot \text{diag}_{\text{matrix}}(\tau)$$

na qual  $\Omega$  é uma matriz de correlação com 1 na sua diagonal e os demais elementos entre -1 e 1  $\rho \in (-1, 1)$ .  $\tau$  é um vetor composto pelas variâncias das variáveis de  $\Sigma$  (a diagonal de  $\Sigma$ ).

# Prioris para Matrizes de Covariância

Adicionalmente a matriz de correlação  $\Omega$  pode ser decomposta mais uma vez para maior eficiência computacional. Como toda matriz de correlação é simétrica e definitiva positiva (todos seus autovalores são números reais  $\mathbb{R}$  e positivos  $> 0$ ), podemos usar a Decomposição Cholesky para decompô-la em uma matriz triangular (que é muito mais eficiente computacionalmente):

$$\Omega = \mathbf{L}_\Omega \mathbf{L}_\Omega^T$$

onde  $\mathbf{L}_\Omega$  é uma matriz triangular.

O que falta é definirmos então uma *priori* para a matriz de correlação  $\Omega$ . Até pouco tempo atrás, usávamos uma distribuição de Wishart como *priori*(Gelman et al., 2013b). Mas essa prática foi abandonada após a proposição da distribuição LKJ de Lewandowski et al. (2009) (LKJ são os nomes dos autores – Lewandowski, Kurowicka e Joe) como *priori* de matrizes de correlação.

## *Random-intercept-slope model*

A terceira abordagem é o *random-intercept-slope model* na qual especificamos para cada grupo uma constante diferente juntamente com coeficientes diferentes para cada variável independente desejada. É claro também resulta em uma constante global. Essas constantes e coeficientes à nível de grupo são amostrados de duas ou mais *hiperprioris*.

No caso de *random-intercept-slope model*, a formula a ser usada segue este padrão:

$$y \sim (1 + x_1 | \text{group}) + (1 + x_2 | \text{group})$$

# Especificação Matemática – *Random-intercept-slope model*

$\mathbf{y} \sim \text{Normal}(\alpha + \alpha_j + \mathbf{X} \cdot \boldsymbol{\beta}_j, \sigma)$

$\alpha \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$

$\alpha_j \sim \text{Normal}(0, \tau)$

$\boldsymbol{\beta}_j \sim \text{Normal Multivariada}(\boldsymbol{\mu}_j, \Sigma) \quad \text{para } j \in \{1, \dots, J\}$

$\Sigma \sim \text{LKJ}(\eta)$

$\tau \sim \text{Cauchy}^+(0, \psi_\alpha)$

$\sigma \sim \text{Exponential}(\lambda_\sigma)$

# Modelos Multiníveis no rstanarm

```
stan_glmer(  
  y ~ (1 + x1 | group) + (1 + x2 | group)  
  ...  
  prior_intercept = ...,  
  prior_covariance = decov(1) # LKJ com  $\eta = 1$   
)
```

# Modelos Multiníveis no brms

```
brm(  
  y ~ (1 + x1 | group) + (1 + x2 | group)  
  ...  
  # LKJ com  $\eta = 1$   
  prior = c(.., prior(lkj_corr_cholesky(1), class = L))  
)
```

# Sumário para *Markov Chain Monte Carlo* - MCMC

## 11.1 Leituras Recomendadas

## 11.2 Por quê Precisamos de MCMC?

### 11.2.1 Correntes Markov

## 11.3 Algoritmos de MCMC

### 11.3.1 Metropolis

### 11.3.2 Metropolis-Hastings

### 11.3.3 Limitações dos Algoritmos Metropolis

### 11.3.4 Gibbs

### 11.3.5 Limitações do Algoritmo de Gibbs

### 11.3.6 Hamiltonian Monte Carlo (HMC)

### 11.3.7 No-U-Turn-Sampler (NUTS)

### 11.3.8 Limitações de HMC e NUTS

## 11.4 Convergência de Correntes Markov

### 11.4.1 Métricas de Convergência

### 11.4.2 Visualizações de Convergência

### 11.4.3 O que fazer se Correntes Markov não Convergirem

# *Markov Chain Monte Carlo - MCMC - Leituras Recomendadas*

- Gelman et al. (2013b)
  - Capítulo 10: Introduction to Bayesian computation
  - Capítulo 11: Basics of Markov chain simulation
  - Capítulo 12: Computationally efficient Markov chain simulation
- McElreath (2020) - Capítulo 9: Markov Chain Monte Carlo
- Neal (2011)
- Betancourt (2017)
- Gelman, Hill e Vehtari (2020) - Capítulo 22, Seção 22.8: Computational efficiency
- Storopoli (2021) - Markov Chain Monte Carlo
- Chib e Greenberg (1995)
- Casella e George (1992)

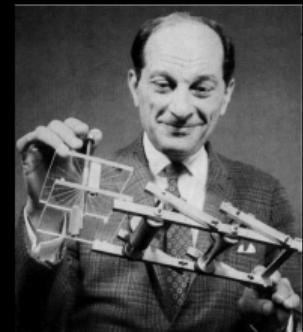
# Métodos de Monte Carlo

- Stan é uma homenagem ao matemático Stanislaw Ulam, que participou do projeto Manhattan e ao tentar calcular o processo de difusão de neutrons para a bomba de hidrogênio acabou criando uma classe de métodos chamada **Monte Carlo** (Eckhardt, 1987).
- Métodos de Monte Carlo possuem como conceito subjacente o uso a aleatoriedade para resolver problemas que podem ser determinísticos em princípio. Eles são freqüentemente usados em problemas físicos e matemáticos e são mais úteis quando é difícil ou impossível usar outras abordagens.



# História dos Métodos de Monte Carlo<sup>71</sup>

- A ideia do método veio enquanto jogava paciência durante sua recuperação de uma cirurgia, Ulam pensou em jogar centenas de jogos para estimar estatisticamente a probabilidade de um resultado bem-sucedido
- Ulam descreveu a ideia para John von Neumann em 1946
- Por ser secreto, o trabalho de von Neumann e Ulam exigia um codinome. Um colega de von Neumann e Ulam, Nicholas Metropolis, sugeriu usar o nome Monte Carlo, que se refere ao Casino Monte Carlo em Mônaco, onde o tio de Ulam (Michał Ulam) pedia dinheiro emprestado a parentes para jogar.



<sup>71</sup>para quem se interessou, a história se encontra em Eckhardt (1987)

# Por quê Precisamos de MCMC?

A principal barreira computacional para estatística Bayesiana é o denominador  $P(\text{data})$  da fórmula de Bayes:

$$P(\theta \mid \text{data}) = \frac{P(\theta) \cdot P(\text{data} \mid \theta)}{P(\text{data})}$$

Em casos discretos podemos fazer o denominador virar a soma de todos os parâmetros usando a **regra da cadeia de probabilidade** (*chain rule*):

$$P(A, B \mid C) = P(A \mid B, C) \times P(B \mid C)$$

Isto também é chamado de **marginalização**:

$$P(\text{data}) = \sum_{\theta} P(\text{data} \mid \theta) \times P(\theta)$$

# Por quê Precisamos de MCMC?

Porém no caso de valores contínuos o denominador  $P(\text{data})$  vira uma integral bem grande e complicada de calcular:

$$P(\text{data}) = \int_{\theta} P(\text{data} | \theta) \times P(\theta) d\theta$$

Em muitos casos essa integral vira *intratável* (incalculável) e portanto devemos achar outras maneiras de calcular a probabilidade posterior  $P(\theta | \text{data})$  de Bayes sem usar o denominador  $P(\text{data})$ .

**É aqui que entra Métodos de Monte Carlo!**

# Para quê serve o denominador $P(\text{data})$

Para normalizar a posterior com o intuito de torná-la uma distribuição probabilística válida. Isto quer dizer que a soma de todas as probabilidades dos eventos possíveis da distribuição devem ser iguais a 1:

- no caso de distribuição probabilística **discreta**:

$$\sum_{\theta} P(\theta \mid \text{data}) = 1$$

- no caso de distribuição probabilística **contínua**:

$$\int_{\theta} P(\theta \mid \text{data}) d\theta = 1$$

# Se removermos o denominador de Bayes o que temos?

Ao removermos o denominador (data) temos que a posterior  $P(\theta | \text{data})$  é **proporcional** à *priori* multiplicada pela verossimilhança  $P(\theta) \cdot P(\text{data} | \theta)$ :

$$P(\theta | \text{data}) \propto P(\theta) \cdot P(\text{data} | \theta)$$

# Método de Montecarlo com Correntes Markov – (MCMC)

Aí que entra **Método Montecarlo com Correntes Markov**<sup>72</sup>.

MCMC é uma classe ampla de ferramentas computacionais para aproximação de integrais e geração de amostras de uma probabilidade posterior (Brooks et al., 2011).

MCMC é usada quando não é possível coletar amostras de  $\theta$  direto da distribuição probabilística posterior  $P(\theta | \text{data})$ . Ao invés disso, nos coletamos amostras de maneira iterativa que a cada passo do processo nós esperamos que a distribuição da qual amostramos  $P^*(\theta^{(*)} | \text{data})$  se torna cada vez mais similar à posterior  $P(\theta | \text{data})$ .

Tudo isso é para **eliminar o cálculo** (muitas vezes impossível) do **denominador**  $P(\text{data})$ .

---

<sup>72</sup>do inglês *Markov Chain Monte Carlo* (MCMC)

# Correntes Markov

- A ideia é **definir uma corrente Markov ergódica** (quer dizer que há uma distribuição estacionária única) dos quais o conjunto de estados possíveis é o espaço amostral e a distribuição estacionária é a distribuição a ser *aproximada* (ou *amostrada*).
- Seja  $X_0, X_1, \dots, X_n$  uma simulação da corrente. A corrente Markov **converge à distribuição estacionária de qualquer estado inicial**  $X_0$  após um **número suficiente grande de iterações**  $r$ , a distribuição do estado  $X_r$  estará similar à distribuição estacionária, então podemos usá-la com amostra.



# Correntes Markov

- As correntes Markov possuem uma propriedade que a distribuição probabilística do próximo estado depende **apenas do estado atual e não na sequência de eventos que precederam:**

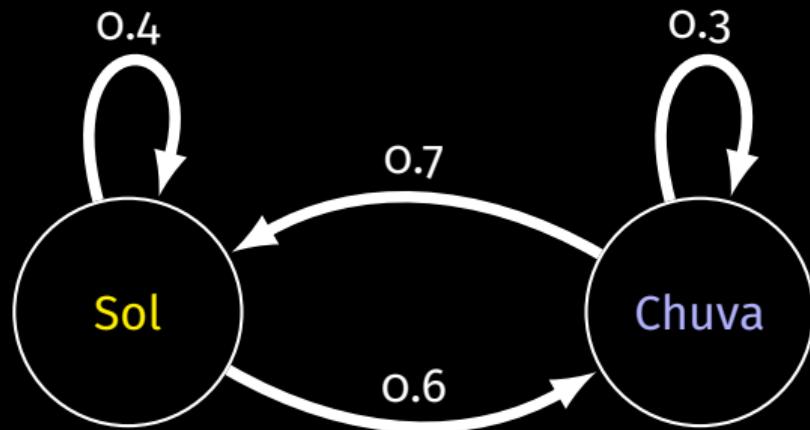
$$P(X_{n+1} = x \mid X_0, X_1, X_2, \dots, X_n) = P(X_{n+1} = x \mid X_n).$$

Essa propriedade é chamada de **Markoviana**.

- Similarmente, repetindo esse argumento com  $X_r$  como o ponto inicial, podemos usar  $X_{2r}$  como amostra, e assim por diante. Podemos então usar a sequência de estados  $X_r, X_{2r}, X_{3r}, \dots$  como quase **amostras independentes** da distribuição estacionária da corrente Markov.



# Exemplo de Corrente Markov



# Correntes Markov

A eficácia dessa abordagem depende em:

- **o quanto grande  $r$  deve ser** para garantir uma **amostra adequadamente boa**
- **poder computacional** requerido para cada iteração da corrente Markov.

Além disso, é costumeiro descartarmos as primeiras iterações do algoritmo pois elas costumam não ser representativas da distribuição a ser aproximada. Nas iterações iniciais de algoritmos MCMC geralmente a corrente Markov está em um processo de aquecimento<sup>73</sup> (*warm-up*) e seu estado está bem distante do ideal para começarmos uma amostragem fidedigna.

Geralmente, recomenda-se que se descarte metade das iterações (Gelman et al., 2013a).

---

<sup>73</sup>Algumas referências chamam esse processo de *burnin*

# Algoritmos de MCMC

Temos **MUITOS** algoritmos de MCMC<sup>74</sup> Mas aqui vamos cobrir duas classes de algoritmos MCMC:

- Metropolis-Hastings (Hastings, 1970; Metropolis et al., 1953)
- Hamiltonian Monte Carlo<sup>75</sup> (Betancourt, 2017; Neal, 2011)

---

<sup>74</sup>Veja a página da Wikipedia para uma listagem completa

<sup>75</sup>às vezes chamado de *Hybrid Monte Carlo*, especialmente na literatura de Física

# Classe de Algoritmos MCMC – Metropolis-Hastings

Os primeiros algoritmos de MCMC. Usam uma regra de aceitação/rejeição das propostas. Caracterizados por propostas oriundas de um passeio aleatório<sup>76</sup> no espaço amostral. O algoritmo de **Gibbs** pode ser visto como um **caso especial** do algoritmo de MH porque todas as propostas são aceitas (Gelman, 1992)

Assintoticamente, possuem uma taxa de aceitação de 23.4% e o custo de cada iteração é  $\mathcal{O}(d)$ , na qual  $d$  é a dimensão do espaço amostral (Beskos et al., 2013).

---

<sup>76</sup>random walk

# Classe de Algoritmos MCMC – Hamiltonian Monte Carlo

Os algoritmos MCMC mais eficientes na atualidade. Tenta evitar o comportamento de passeio aleatório introduzindo um vetor de momento auxiliar e implementando dinâmicas Hamiltonianas. As propostas são "guiadas" para regiões de maior densidade do espaço amostral. Isso faz com que HMC seja **ordens de magnitude mais eficiente que MH e Gibbs.**

Assintoticamente, possuem uma taxa de aceitação de 65.1% e o custo de cada iteração é  $\mathcal{O}(d^{\frac{1}{4}})$ , na qual  $d$  é a dimensão do espaço amostral (Beskos et al., 2013).

# Algoritmo de Metropolis

O primeiro algoritmo MCMC amplamente utilizado para gerar amostras de correntes Markov foi originário na física na década de 1950 e chama-se Metropolis (Metropolis et al., 1953) em homenagem ao primeiro autor Nicholas Metropolis.

Em síntese, o algoritmo de Metropolis é uma adaptação de um passeio aleatório com uma regra de aceitação/rejeição para convergir à distribuição-alvo.

O algoritmo de Metropolis usa uma **distribuição de propostas**  $J_t(\theta^{(*)})$  para definir próximos valores da distribuição  $P^*(\theta^{(*)} | \text{data})$ . Essa distribuição deve ser simétrica:

$$J_t(\theta^{(*)} | \theta^{(t-1)}) = J_t(\theta^{(t-1)} | \theta^{(*)})$$



# Algoritmo de Metropolis

A essência do algoritmo é um passeio aleatório pelo espaço amostral dos parâmetros, onde a probabilidade da corrente Markov mudar de estado é definida como:

$$P_{\text{mudar}} = \min \left( \frac{P(\boldsymbol{\theta}_{\text{proposto}})}{P(\boldsymbol{\theta}_{\text{atual}})}, 1 \right).$$

Isso quer dizer a corrente Markov somente mudará para um novo estado em duas condições:

- Quando a probabilidade dos parâmetros propostos pelo passeio aleatório  $P(\boldsymbol{\theta}_{\text{proposto}})$  é **maior** que a probabilidade dos parâmetros do estado atual  $P(\boldsymbol{\theta}_{\text{atual}})$ , mudamos com 100% de probabilidade.
- Quando a probabilidade dos parâmetros propostos pelo passeio aleatório  $P(\boldsymbol{\theta}_{\text{proposto}})$  é **menor** que a probabilidade dos parâmetros do estado atual  $P(\boldsymbol{\theta}_{\text{atual}})$ , mudamos com probabilidade igual a proporção dessa diferença.

# Algoritmo de Metropolis

## Algoritmo 1: Metropolis

Defina um ponto inicial  $\theta^{(0)} \in \mathbb{R}^p$  do qual  $P(\theta^{(0)} | \mathbf{y}) > 0$

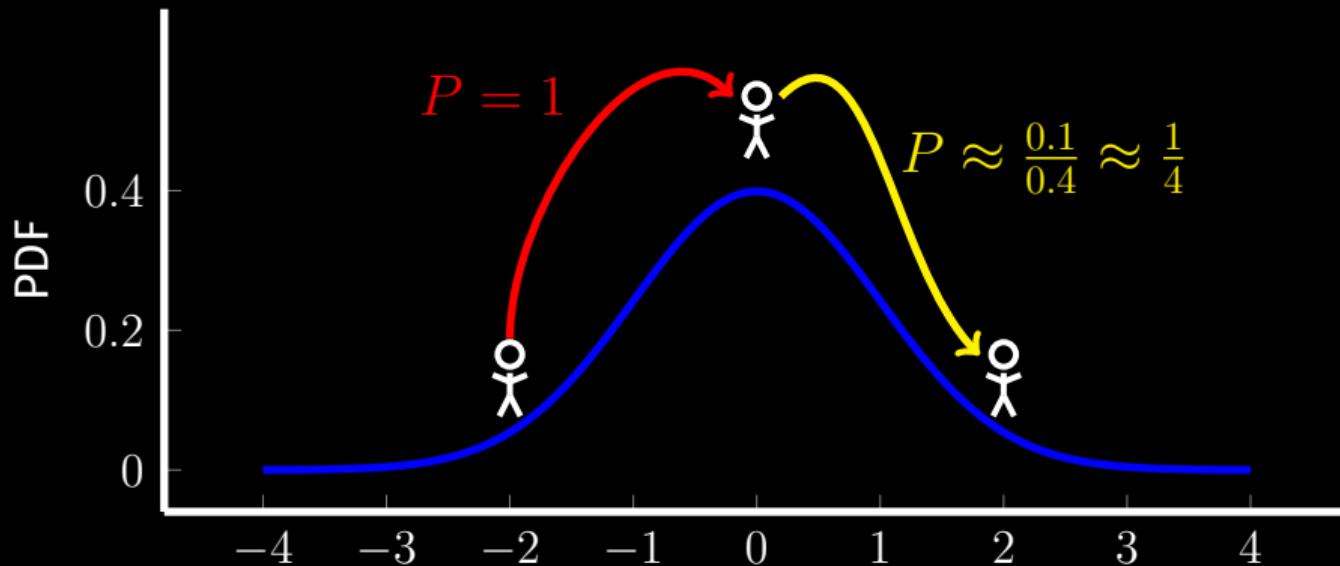
para  $t = 1, 2, \dots$

Amostra uma proposta  $\theta^{(*)}$  de uma distribuição de propostas no tempo  $t$ ,  $J_t(\theta^{(*)} | \theta^{(t-1)})$

Como regra de aceitação/rejeição calcule a proporção das probabilidades:  $r = \frac{P(\theta^{(*)} | \mathbf{y})}{P(\theta^{(t-1)} | \mathbf{y})}$

Designe:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{com probabilidade } \min(r, 1) \\ \theta^{(t-1)} & \text{caso contrário} \end{cases}$

# Intuição Visual de Metropolis



# Algoritmo de Metropolis

Na década de 1970, surgiu um generalização do algoritmo de Metropolis que **não** necessita que as distribuições de proposta sejam simétricas:

$$J_t(\boldsymbol{\theta}^{(*)} \mid \boldsymbol{\theta}^{(t-1)}) \neq J_t(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^{(*)})$$

A generalização foi proposta por Wilfred Keith Hastings (Hastings, 1970) e chama-se algoritmo de **Metropolis-Hastings**.



# Algoritmo de Metropolis - Hastings

## Algoritmo 2: Metropolis-Hastings

Defina um ponto inicial  $\theta^{(0)} \in \mathbb{R}^p$  do qual  $P(\theta^{(0)} | \mathbf{y}) > 0$

para  $t = 1, 2, \dots$

Amostra uma proposta  $\theta^{(*)}$  de uma distribuição de propostas no tempo  $t$ ,  
 $J_t(\theta^{(*)} | \theta^{(t-1)})$

Como regra de aceitação/rejeição calcule a proporção das probabilidades:

$$r = \frac{\frac{P(\theta^{(*)} | \mathbf{y})}{J_t(\theta^{(*)} | \theta^{(t-1)})}}{\frac{P(\theta^{(t-1)} | \mathbf{y})}{J_t(\theta^{(t-1)} | \theta^{(*)})}}$$

Designe:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{com probabilidade } \min(r, 1) \\ \theta^{(t-1)} & \text{caso contrário} \end{cases}$

# Animação Metropolis<sup>77</sup>

## Animação Metropolis

---

<sup>77</sup>veja Metropolis em ação no chi-feng/mcmc-demo

# Limitações dos Algoritmos Metropolis

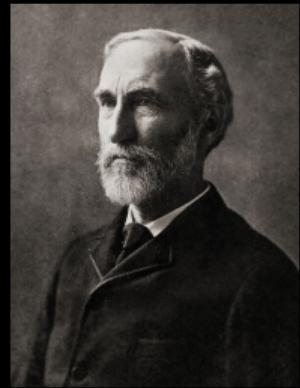
As limitações do algoritmo de Metropolis-Hastings são principalmente **computacionais**:

- Com propostas geradas aleatoriamente, geralmente leva um grande número de iterações para entrar em áreas de densidade posterior mais alta (mais provável).
- Mesmo algoritmos de Metropolis-Hastings eficientes às vezes aceitam menos de 25% das propostas (Beskos et al., 2013; Roberts et al., 1997).
- Em situações dimensionais mais baixas, o poder computacional aumentado pode compensar a eficiência mais baixa até certo ponto. Mas em situações de modelagem de dimensões mais altas e mais complexas, computadores maiores e mais rápidos sozinhos raramente são suficientes para superar o desafio.

# Algoritmo de Gibbs

Para contornar o problema de baixa taxa de aceitação dos algoritmos de Metropolis foi desenvolvido o algoritmo de Gibbs que **não possui uma regra de aceitação/rejeição** para a mudança de estado da corrente Markov: **Todas as propostas são aceitas!**

O algoritmo de Gibbs teve ideia original concebida pelo físico Josiah Willard Gibbs em referência a uma analogia entre um algoritmo de amostragem e a física estatística (*statistical physics* um ramo da física que tem sua base em mecânica estatística, *statistical mechanics*). O algoritmo foi descrito pelos irmãos Stuart e Donald Geman em 1984 (Geman & Geman, 1984), cerca de oito décadas após a morte de Gibbs.



# Algoritmo de Gibbs

O algoritmo de Gibbs é muito útil em espaços amostrais multidimensionais<sup>78</sup>. Também é conhecido como amostragem condicional alternativa (*alternating conditional sampling*), pois amostramos sempre um parâmetro **condicionado** à probabilidade dos outros parâmetros do modelo.

O algoritmo de Gibbs pode ser visto como um **caso especial** do algoritmo de Metropolis-Hastings porque todas as propostas são aceitas (Gelman, 1992).

A essência do algoritmo de Gibbs é a amostragem de parâmetros condicionada à outros parâmetros:

$$P(\theta_1 \mid \theta_2, \dots, \theta_p)$$

---

<sup>78</sup>no qual há bem mais que 2 parâmetros a serem amostrados da probabilidade posterior

# Algoritmo de Gibbs

## Algoritmo 3: Gibbs

Defina um ponto inicial  $\theta^{(0)} \in \mathbb{R}^p$  do qual  $P(\theta^{(0)} | \mathbf{y}) > 0$

**para**  $t = 1, 2, \dots$

Designe:  $\theta^{(t)} = \begin{cases} \theta_1^{(t)} & \sim P\left(\theta_1 | \theta_2^{(0)}, \dots, \theta_p^{(0)}\right) \\ \theta_2^{(t)} & \sim P\left(\theta_2 | \theta_1^{(t-1)}, \dots, \theta_p^{(0)}\right) \\ \vdots \\ \theta_p^{(t)} & \sim P\left(\theta_p | \theta_1^{(t-1)}, \dots, \theta_{p-1}^{(t-1)}\right) \end{cases}$

# Animação Gibbs<sup>79</sup>

## Animação Gibbs

---

<sup>79</sup>Veja Gibbs em ação no chi-feng/mcmc-demo

# Limitações do Algoritmo de Gibbs

A principal limitação do algoritmo de Gibbs é com relação a **amostragem condicional alternativa**:

- Em Metropolis temos propostas aleatórias de uma distribuição de propostas na qual amostramos cada parâmetro **incondicionalmente** à outros parâmetros e de maneira **simultânea** usando a probabilidade conjunta desses parâmetros. As mudanças de estado da corrente Markov são então executadas **multidimensionalmente**. Isto provoca movimentos "**diagonais**" multidimensionais.
- No caso do algoritmo de Gibbs essa movimentação se dá apenas em um único parâmetro, pois amostramos **sequencialmente** e **condicionalmente** à outros parâmetros. Isto provoca movimentos **horizontais/verticais** unidimensionais, mas nunca movimentos diagonais multidimensionais.

# Classe de Algoritmos MCMC - Hamiltonian Monte Carlo (HMC)

Os problemas de baixas taxas de aceitação de propostas das técnicas de Metropolis e do desempenho baixo do algoritmo de Gibbs em problemas multidimensionais nas quais a geometria da posterior é complexa fizeram com que surgisse uma nova técnica MCMC usando dinâmica Hamiltoniana (em homenagem ao físico irlandês William Rowan Hamilton.



# Algoritmo de HMC

O HMC é uma adaptação da técnica de Metropolis e emprega um esquema guiado de geração de novas proposta: isso melhora a taxa de aceitação de propostas e, consequentemente, a eficiência.

Mais especificamente, o HMC usa o gradiente do log da posterior para direcionar a cadeia de Markov para regiões de maior densidade posterior, onde a maioria das amostras são coletadas.

$$\frac{d \log P(\boldsymbol{\theta} | \mathbf{y})}{d\theta}$$

Como resultado, uma corrente Markov com o algoritmo HMC bem ajustada aceitará propostas em uma taxa muito mais alta do que o algoritmo Metropolis tradicional (Beskos et al., 2013; Roberts et al., 1997).

# História do Algoritmo de HMC

HMC foi inicialmente descrito na literatura de física<sup>80</sup> (Duane et al., 1987).

Logo depois, HMC foi aplicado a problemas estatísticos por Neal (1994) que chamou de *Hamiltonian Monte Carlo* – HMC).

Para uma discussão aprofundada (que não é o foco deste conteúdo) de HMC eu recomendo Neal (2011) e Betancourt (2017).

---

<sup>80</sup>que chamaram de "Hybrid" Monte Carlo – HMC

# O que muda com HMC?

HMC usa dinâmica Hamiltoniana aplicada para partículas explorando de maneira mais eficiente a geometria de uma probabilidade posterior.

Além de explorar melhor a geometria da posterior e tolerar geometrias complexas, HMC é muito mais eficiente que Metropolis e não sofre do problema de correlação dos parâmetros que Gibbs.

# Intuição por trás do Algoritmo de HMC

Para cada componente  $\theta_j$ , o HMC adiciona uma variável de momento  $\phi_j$ . A densidade posterior  $P(\boldsymbol{\theta} | y)$  é incrementada por uma distribuição independente  $P(\boldsymbol{\phi})$  dos momentos, definindo assim uma distribuição conjunta:

$$P(\boldsymbol{\theta}, \boldsymbol{\phi} | y) = P(\boldsymbol{\phi}) \cdot P(\boldsymbol{\theta} | y)$$

O HMC usa uma distribuição de propostas que muda dependendo do estado atual na corrente Markov. O HMC descobre a direção em que a distribuição posterior aumenta, chamada de *gradiente*, e distorce a distribuição de propostas em direção ao *gradiente*.

A probabilidade da corrente Markov mudar de estado no algoritmo HMC é definida como:

$$P_{\text{mudar}} = \min \left( \frac{P(\boldsymbol{\theta}_{\text{proposto}}) \cdot P(\boldsymbol{\phi}_{\text{proposto}})}{P(\boldsymbol{\theta}_{\text{atual}}) \cdot P(\boldsymbol{\phi}_{\text{atual}})}, 1 \right)$$

# Distribuição dos Momentos – $P(\phi)$

Normalmente damos a  $\phi$  uma distribuição normal multivariada com média  $\boldsymbol{\mu}$  e covariância de  $\mathbf{M}$ , uma "matriz de massa".

Para manter as coisas um pouco mais simples, usamos uma matriz de massa diagonal  $\mathbf{M}$ . Isso faz com que os componentes de  $\phi$  sejam independentes com

$$\phi_j \sim \text{Normal}(0, M_{jj})$$

# Algoritmo de HMC

## Algoritmo 4: Hamiltonian Monte Carlo (HMC)

Defina um ponto inicial  $\theta^{(0)} \in \mathbb{R}^p$  do qual  $P(\theta^{(0)} | \mathbf{y}) > 0$

Amostre  $\phi$  de uma  $\text{Normal}(\mathbf{0}, \mathbf{M})$

Simultaneamente amostre  $\theta^{(*)}$  e  $\phi$  com  $L$  passos e tamanho de passo  $\epsilon$ .

Defina o valor atual  $\theta$  como valor proposto  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta$

**para**  $1, 2, \dots, L$

Use o gradiente do log da posterior de  $\theta^{(*)}$  para produzir um meio-passo de  $\phi$ :

$$\phi \leftarrow \phi + \frac{1}{2}\epsilon \frac{d \log P(\theta^{(*)} | \mathbf{y})}{d\theta}$$

Use  $\phi$  para atualizar  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta^{(*)} + \epsilon \mathbf{M}^{-1} \phi$

Novamente use o gradiente de  $\theta$  para produzir um meio-passo de  $\phi$ :  $\phi \leftarrow \phi + \frac{1}{2}\epsilon \frac{d \log P(\theta^{(*)} | \mathbf{y})}{d\theta}$

Como regra de aceitação/rejeição calcule:  $r = \frac{P(\theta^{(*)} | \mathbf{y}) P(\phi^{(*)})}{P(\theta^{(t-1)} | \mathbf{y}) P(\phi^{(t-1)})}$

Designe:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{com probabilidade } \min(r, 1) \\ \theta^{(t-1)} & \text{caso contrário} \end{cases}$

# Animação HMC<sup>81</sup>

## Animação HMC

---

<sup>81</sup>Veja HMC em ação no [chi-feng/mcmc-demo](#)

# Um interlúdio de Integrador Numéricos

No campo das equações diferenciais ordinais temos a ideia de discretizar um sistema de equações diferenciais ordinais ao aplicar um pequeno passo  $\epsilon^{82}$ . Tais abordagem são chamadas de **integradores numéricos** e comportam uma **ampla classe** de ferramentas.

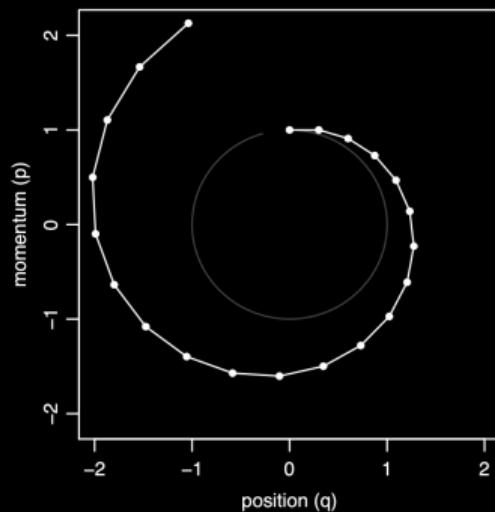
O mais famoso e simples desses integradores numéricos é o método de Euler. No qual usa-se um tamamho de passo  $\epsilon$  para calcular a solução numérica do estado em um futuro tempo  $t$  a partir de condições iniciais específicas.

---

<sup>82</sup> algumas vezes também chamado de  $h$

# Um interlúdio de Integrador Numéricos

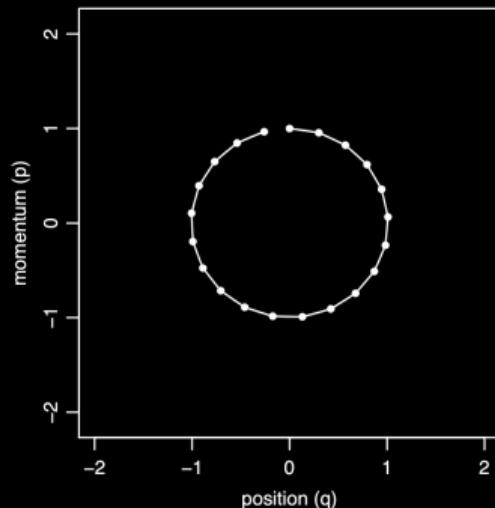
O problema é que o método de Euler quando aplicado para dinâmicas Hamiltonianas é que ele não preserva o volume. Uma das propriedades fundamentais das dinâmicas Hamiltonianas é que elas preservam volume, um resultado chamado de Teorema de Liouville. Isto faz com que o método de Euler seja uma péssima escolha como integrador numéricico de um algoritmo HMC.



Método de Euler num algoritmo HMC com  $\epsilon = 0.3$  e  $L = 20$

# Um interlúdio de Integrador Numéricos<sup>83</sup>

Para preservação de volumes precisamos usar um **integrador simplético**. Integradores simpléticos são no máximo de ordem 2 e precisam ser usados com um tamanho de passo  $\epsilon$  constante. Um dos principais integradores numéricos simpléticos usado em dinânicas Hamiltonianas é o integrador **Störmer–Verlet**, também conhecido como *leapfrog*.

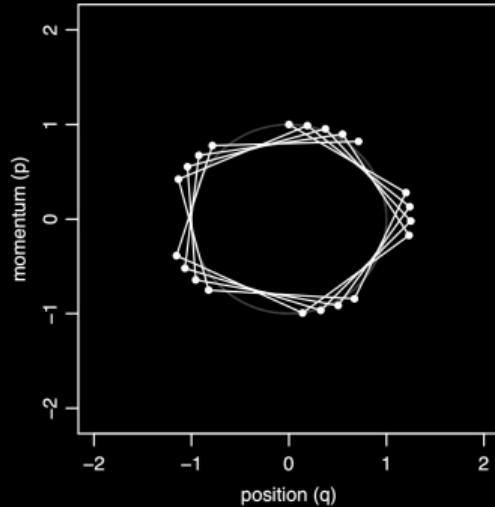


Integrador *Leapfrog* num algoritmo HMC com  $\epsilon = 0.3$  e  $L = 20$

<sup>83</sup>Um excelente livro-texto para integradores numéricos e integradores simpléticos é Iserles (2008)

# Limitações do Algoritmo HMC

Como vocês podem ver o algoritmo de HMC é muito sensível a escolha da quantidade de passos  $L$  e do tamanho do passo  $\epsilon$ . Em especial o integrador *leapfrog* permite apenas um  $\epsilon$  constante, portanto temos um equilíbrio delicado entre  $L$  e  $\epsilon$ . Em termos algorítmicos,  $L$  e  $\epsilon$  são hiperparâmetros (tem que ser cuidadosamente ajustados).



Integrador *Leapfrog* num algoritmo HMC com  $\epsilon = 1.2$  e  $L = 20$

## No-U-Turn-Sampler (NUTS)

Em HMC, conseguimos ajustar o  $\epsilon$  durante a execução do algoritmo. Mas, geralmente precisamos executar algumas vezes o amostrador HMC para ajustar o  $L$ .

Aqui vem a ideia do **No-U-Turn-Sampler (NUTS)** (Hoffman & Gelman, 2011). Não é preciso ajustar **nada** apenas "apertar" o botão. Ele calcula automaticamente  $\epsilon$  e  $L$ .

## No-U-Turn-Sampler (NUTS)

Mais especificamente precisamos de um critério que informe que já simulamos as dinâmicas Hamiltonianas por "tempo suficiente". *i.e.* simular as dinâmicas por mais tempo não aumentaria a distância entre a proposta  $\theta^{(*)}$  e o valor atual  $\theta$ .

NUTS então usa um critério baseado no produto interno entre os vetores do momento atual  $\phi$  e a diferença entre os vetores da proposta  $\theta^{(*)}$  e o valor atual  $\theta$ , que é a derivada com respeito ao tempo  $t$  de metade da distância ao quadrado entre  $\theta$  e  $\theta^{(*)}$

$$(\theta^{(*)} - \theta) \cdot \phi = (\theta^{(*)} - \theta) \cdot \frac{d}{dt}(\theta^{(*)} - \theta) = \frac{d}{dt} \frac{(\theta^{(*)} - \theta) \cdot (\theta^{(*)} - \theta)}{2}$$

## No-U-Turn-Sampler (NUTS)

Isso sugere um algoritmo que não permite com que as propostas sejam guiadas de maneira infinita até que a distância entre a proposta  $\theta^{(*)}$  e o valor atual  $\theta$  seja menor que zero.

Isto quer dizer que tal algoritmo não **permitirá meia-voltas** (*u-turns*).

# No-U-Turn-Sampler (NUTS)

NUTS usa o integrador *leapfrog* para criar uma árvore binária da qual os nós-folha são as posições do momento  $\phi$  traçando tanto um caminho para frente ( $t + 1$ ) quanto para trás ( $t - 1$ ) em um tempo fictício em um determinado tempo  $t$ . O crescimento dos nós-folha são **interrompidos** quando é detectado meia-volta tanto para frente quanto para trás.



## No-U-Turn-Sampler (NUTS)

NUTS também um procedimento chamado *Dual Averaging* (Nesterov, 2009) para ajustar simultaneamente  $\epsilon$  e  $L$  ao considerar o produto  $\epsilon \cdot L$ .

Tal ajuste é feito durante a fase de *warmup* e os valores definidos de  $\epsilon$  e  $L$  são mantidos fixos durante a fase de amostragem.

# Algoritmo de NUTS

## Algoritmo 5: No-U-Turn-Sampler (NUTS)

Defina um ponto inicial  $\theta^{(0)} \in \mathbb{R}^P$  do qual  $P(\theta^{(0)} | \mathbf{y}) > 0$

Inicie uma árvore binária vazia com  $2^L$  nós

Amostre  $\phi$  de uma Normal(0, M)

Simultaneamente amostre  $\theta$  e  $\phi$  com L passos e tamanho de passo  $\epsilon$ .

Defina o valor atual  $\theta$  como valor proposto  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta$

para  $1, 2, \dots, 2L$

Escolha uma direção  $v \sim \text{Uniforme}(\{-1, 1\})$

Use o gradiente do log da posterior de  $\theta^{(*)}$  para produzir um meio-passo de  $\phi$  na direção  $v$ :  $\phi \leftarrow \phi + v \frac{1}{2} \epsilon \frac{d \log P(\theta^{(*)} | \mathbf{y})}{d\theta}$

Use  $\phi$  para atualizar  $\theta^{(*)}$ :  $\theta^{(*)} \leftarrow \theta^{(*)} + \epsilon M^{-1} \phi$

Novamente use o gradiente de  $\theta^{(*)}$  para produzir um meio-passo de  $\phi$  na direção  $v$ :  $\phi \leftarrow \phi + v \frac{1}{2} \epsilon \frac{d \log P(\theta^{(*)} | \mathbf{y})}{d\theta}$

Defina o nó  $L_t^v$  como a proposta  $\theta$

se A diferença entre os vetores da proposta  $\theta^{(*)}$  e o valor atual  $\theta$  na direção  $v$  for menor que zero:  $v \frac{d}{dt} \frac{(\theta^{(*)} - \theta^{(*)}) \cdot (\theta^{(*)} - \theta^{(*)})}{2} < 0$

então

| Pare a amostragem de  $\theta^{(*)}$  na direção  $v$  e continue apenas amostrando na direção  $-v$

senão

| se A distância entre os vetores da proposta  $\theta^{(*)}$  e o valor atual  $\theta$  na direção restante  $-v$  for menor que zero:  $-v \frac{d}{dt} \frac{(\theta^{(*)} - \theta^{(*)}) \cdot (\theta^{(*)} - \theta^{(*)})}{2} < 0$

então

| Pare a amostragem de  $\theta^{(*)}$

Como regra de aceitação/rejeição calcule:  $r = \frac{P(\theta^{(*)} | \mathbf{y}) P(\phi^{(*)})}{P(\theta^{(t-1)} | \mathbf{y}) P(\phi^{(t-1)})}$

Designe:  $\theta^{(t)} = \begin{cases} \theta^{(*)} & \text{com probabilidade } \min(r, 1) \\ \theta^{(t-1)} & \text{caso contrário} \end{cases}$

# Animação NUTS<sup>84</sup>

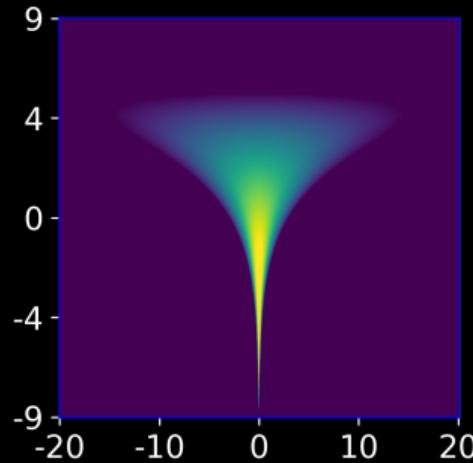
## Animação NUTS

---

<sup>84</sup>Veja NUTS em ação no [chi-feng/mcmc-demo](#)

# Limitações do Algoritmo HMC e NUTS - Funil de Neal (2003)

O famoso funil da morte<sup>85</sup>. Aqui vemos que os algoritmos HMC e NUTS, durante a exploração da posterior, tem que a todo momento trocar valores<sup>86</sup> de  $\epsilon$  e  $L$ .



<sup>85</sup>muito comum em modelos hierárquicos

<sup>86</sup>lembre-se que  $\epsilon$  e  $L$  são definidos na fase de *warmup* e mantidos fixos durante a fase de amostragem

# Funil de Neal (2003) e Parametrização Não-Centralizada<sup>87</sup>

O funil ocorre quando temos uma variável que a sua variância depende da variância de outra em uma escala exponencial. Um exemplo canônico de uma parametrização centralizada é:

$$P(y, x) = \text{Normal}(y | 0, 3) \cdot \text{Normal}\left(x | 0, e^{\left(\frac{y}{2}\right)}\right)$$

Isto ocorre bastante em modelos hierárquicos, na relação dos *prioris* de grupo com a(s) *hiperpriori(s)* global(is). Então, reparametrizamos de maneira não-centrada alterando a geometria da posterior para facilitar a vida do amostrador MCMC:

$$P(\tilde{y}, \tilde{x}) = \text{Normal}(\tilde{y} | 0, 1) \cdot \text{Normal}(\tilde{x} | 0, 1)$$

$$y = \tilde{y} \cdot 3 + 0$$

$$x = \tilde{x} \cdot e^{\left(\frac{y}{2}\right)} + 0$$

---

<sup>87</sup>*Non-Centered Parametrization (NCP)*

# Stan e NUTS

Stan foi o primeiro amostrador MCMC a implementar NUTS. Além disso tem uma rotina otimizada automática de ajuste de  $L$  e  $\epsilon$  durante a fase de *warmup*. Possui os seguintes valores como hiperparâmetros padrões do NUTS<sup>88</sup>:

- **Taxa-alvo de aceitação de propostas Metropolis:** `adapt_delta = 0.8`
- **Profundidade máxima de árvore** (em potências de 2):  
`max_treedepth = 10` (quer dizer  $2^{10} = 1024$ )

---

<sup>88</sup>para mais informações sobre como modificar esses hiperparâmetros consulte a Seção 15.2 do Stan Reference Manual

# Convergência de Correntes Markov

MCMC tem uma propriedade interessante que é garantido que **assintoticamente ele convergirá à distribuição-alvo.**

Ou seja, se tivermos todo o tempo do mundo, é garantido que, irrelevante da geometria da distribuição-alvo (posterior), **MCMC irá lhe dar a resposta correta.**

Porém não temos todo o tempo do mundo. Diferentes algoritmos MCMC, como HMC e NUTS, podem reduzir o tempo de amostragem (e *warmup*) necessários para convergência.

# Métricas de Convergência

Temos algumas maneiras de mensurar se as correntes Markov convergiram à distribuição-alvo, i.e. são "confiáveis":

- Número de Amostras Efetivas (*Effective Sample Size* – ESS): uma aproximação do "número de amostras independentes" geradas por uma corrente Markov.
- $\widehat{R}$  (*Rhat*): escala de Redução potencial, uma métrica de mensuração que as correntes Markov se "misturaram", e, potencialmente, convergiram

# Métricas de Convergência - *Effective Sample Size* (Gelman et al., 2013b)

$$\widehat{n}_{\text{eff}} = \frac{mn}{1 + \sum_{t=1}^T \widehat{\rho}_t}$$

Onde:

- $m$ : número de correntes Markov
- $n$ : amostras totais por corrente Markov (descontando *warmup*)
- $\widehat{\rho}_t$ : uma estimativa de autocorrelação

## Métricas de Convergência - *Rhat* (Gelman et al., 2013b)

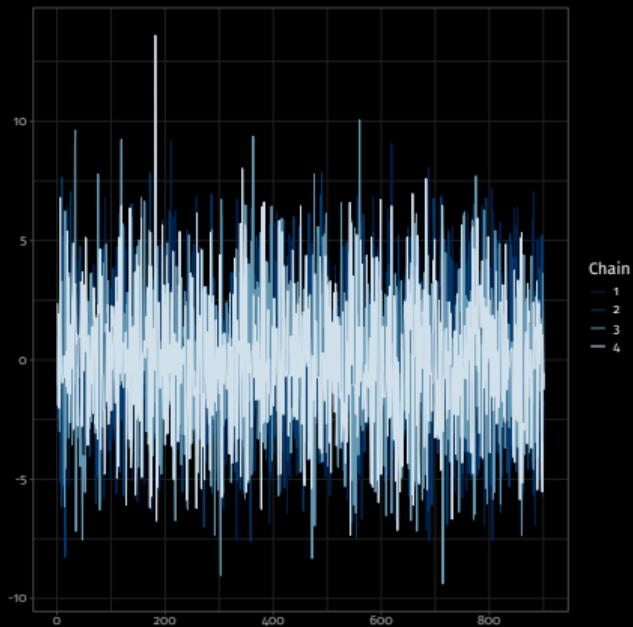
$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^+(\psi | y)}{W}}$$

onde a  $\widehat{\text{var}}^+(\psi | y)$  é a variância das amostras das correntes Markov para um determinado parâmetro  $\psi$  sob uma média ponderada das variâncias intra-correntes (*within-chain*)  $W$  e inter-correntes (*between-chain*)  $B$

$$\widehat{\text{var}}^+(\psi | y) = \frac{n-1}{n}W + \frac{1}{n}B$$

Intuitivamente, seu valor é 1.0 se as correntes estiverem totalmente convergentes. Como uma heurística, se  $\widehat{R}$  for maior que 1.1, você deve se preocupar pois provavelmente as correntes não tenham收敛ido adequadamente.

# Traceplot – Correntes Markov Convergentes



# Rhat – Mensagens de Erro do Stan<sup>89</sup>

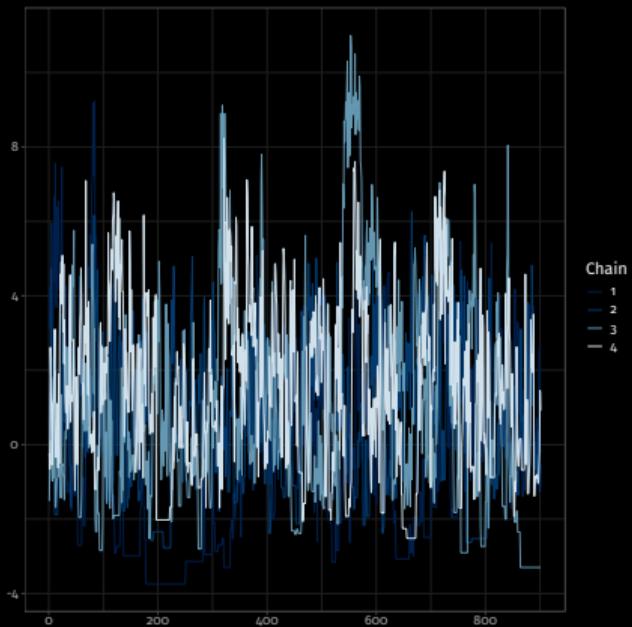
Warning messages:

- 1: There were 275 divergent transitions after warmup. See  
<http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup>  
to **find** out why this **is** a problem and how to eliminate them.
- 2: Examine the **pairs()** **plot** to diagnose sampling problems
  
- 3: The largest **R-hat** is 1.12, indicating chains have not mixed.  
Running the chains **for** more iterations may **help**. See  
<http://mc-stan.org/misc/warnings.html#r-hat>
- 4: Bulk Effective Samples Size (ESS) **is** too low, indicating posterior means and medians may be unreliable.  
Running the chains **for** more iterations may **help**. See  
<http://mc-stan.org/misc/warnings.html#bulk-ess>
- 5: Tail Effective Samples Size (ESS) **is** too low, indicating posterior variances and tail quantiles may be unreliable.  
Running the chains **for** more iterations may **help**. See  
<http://mc-stan.org/misc/warnings.html#tail-ess>

---

<sup>89</sup>além disso não deixe de checar o guia dos **warnings** do Stan

# Traceplot – Correntes Markov Divergentes



# O que fazer se Correntes Markov não Convergirem

**Primeiro:** Antes de fazer ajustes finos no número de correntes chains ou no número de iterações iter (entre outros ...) saiba que o amostrador HMC-NUTS do Stan e seu ecossistema de pacotes (rstanarm e brms inclusos) é **muito eficiente e eficaz em explorar as mais diversas complexas e "malucas" geometrias** de distribuições-alvo posterior.

Os argumentos padrões, iter = 2000, chains = 4 e warmup = floor(iter / 2), funcionam perfeitamente para 99% dos casos (mesmo em modelos complexos).

# O que fazer se Correntes Markov não Convergirem

Dito isto, **na maioria das vezes quando você possui problemas de amostragem e computacionais no seu modelo Bayesiano, o problema está na especificação do modelo e não no algoritmo de amostragem MCMC<sup>90</sup>**

---

<sup>90</sup> Esta frase foi dita por Andrew Gelman (o "pai" do Stan) e é conhecido como o *Folk Theorem* (Gelman, 2008): *"When you have computational problems, often there's a problem with your model"*

# O que fazer se Correntes Markov não Convergirem

Se o seu modelo Bayesiano está com problemas de convergência há alguns passos que podem ser tentados<sup>91</sup>. Aqui listados do mais simples para o mais complexo:

- **Aumentar o número de iterações e correntes:** primeira opção é aumentar o número de iterações do MCMC com o argumento `iter = XXX` e também é possível aumentar o número de correntes com o argumento `chains = X`. Lembrando que o padrão é `iter = 2000` e `chains = 4`.

---

<sup>91</sup>além disso, vale a pena ativar a decomposição QR na matriz  $X$  de dados, criando uma base ortogonal (não correlacionada) para amostragem. Isso faz com a distribuição-alvo (posterior) fique muito mais amigável do ponto de vista topológico/geométrico para o amostrador MCMC explorá-la de maneira mais eficiente e eficaz.

# O que fazer se Correntes Markov não Convergirem

- **Alterar a rotina de adaptação do HMC:** a segunda opção é fazer com que o algoritmo de amostragem HMC fique mais conservador (com proposições de pulos menores). Isto pode ser alterado com o argumento `adapt_delta` da lista de opções `control`. `control = list(adapt_delta = 0.9)`. O padrão do `adapt_delta` é 0.8. Então qualquer valor entre 0.8 e 1.0 o torna mais conservador.
- **Reparametrização do Modelo:** a terceira opção é reparametrizar o modelo. Há duas maneiras de parametrizar o modelo: a primeira com parametrização centrada (*centered parameterization*) e a segunda com parametrização não-centrada (*non-centered parameterization*).

# O que fazer se Correntes Markov não Convergirem

- **Coletar mais dados:** às vezes o modelo é complexo demais e precisamos de uma amostragem maior para conseguirmos estimativas estáveis.
- **Repensar o modelo:** falha de convergência quando temos uma amostragem adequada geralmente é por conta de uma especificação de *prioris* e verossimilhança que não são compatíveis com os dados. Nesse caso, é preciso repensar o processo gerativo de dados no qual os pressupostos do modelo estão ancorados.

# Sumário para Comparação de Modelos

12.1 Leituras Recomendadas

12.2 Por quê Comparar Modelos?

12.3 Técnicas de Comparação de Modelos

12.3.1 Precisão Preditiva

12.3.2 *Leave-One-Out Cross-Validation* (LOO)

12.3.3 *Widely Applicable Information Criteria* (WAIC)

12.3.4 *K-fold Cross-Validation* (K-fold CV)

12.4 *Pareto Smoothed Importance Sampling LOO* (PSIS-LOO)

12.5 Comparação de Modelos no `rstanarm`

12.6 Comparação de Modelos no `brms`

# Comparação de Modelos - Leituras Recomendadas

- Gelman et al. (2013b) - Capítulo 7: Evaluating, comparing, and expanding models
- Gelman, Hill e Vehtari (2020) - Capítulo 11, Seção 11.8: Cross validation
- McElreath (2020) - Capítulo 7, Seção 7.5: Modelcomparison
- Vehtari et al. (2015)
- Tutorial do loo de Vehtari et al. (2020)
- Storopoli (2021) - Comparação de Modelos
- Spiegelhalter et al. (2002)
- Van Der Linde (2005)
- Watanabe e Opper (2010)
- Gelfand (1996)
- Watanabe e Opper (2010)
- Geisser e Eddy (1979)

# Por quê Comparar Modelos?

Depois de estimarmos um modelo Bayesiano, muitas vezes queremos medir sua precisão preditiva, por si só ou para fins de comparação, seleção ou cálculo de média do modelo (Geisser & Eddy, 1979).

# Mas, e as Verificações Preditivas da Posterior?

É uma maneira subjetiva e arbitrária de analisarmos e compararmos modelos entre si usando sua precisão preditiva.

Há uma maneira objetiva de compararmos modelos Bayesianos com uma métrica robusta que nos ajude a selecionar qual o melhor modelo dentre o rol de modelos candidatos.

Ter uma maneira objetiva de comparar modelos e escolher o melhor dentre eles é muito importante pois no *workflow* Bayesiano geralmente temos diversas iterações entre *prioris* e funções de verossimilhança o que ocasiona na criação de diversos modelos diferentes (Gelman, Vehtari et al., 2020).

# Técnicas de Comparação de Modelos

Temos diversas técnicas de comparação de modelos que usam a precisão preditiva, sendo as principais:

- *Leave-one-out cross-validation* (LOO) (Vehtari et al., 2015)
- *Deviance Information Criterion* (DIC) (Spiegelhalter et al., 2002), mas sabe-se que tem alguns problemas, que surgem em parte por não ser totalmente Bayesiano, pois se baseia em uma estimativa pontual (Van Der Linde, 2005)
- *Widely Applicable Information Criteria* (WAIC) (Watanabe & Opper, 2010), totalmente Bayesiano no sentido de que usa toda a distribuição posterior, e é assintoticamente igual ao LOO (Vehtari et al., 2015)

# Interlúdio Histórico

Antigamente, não havia esse poder computacional e abundância de dados. Comparação de modelos eram baseados em uma métrica de divergência teórica oriunda da entropia da teoria da informação:

$$H(p) = -\text{E} \log(p_i) = -\sum_{i=1}^N p_i \log(p_i)$$

Calculamos a divergência<sup>92</sup> multiplicando por  $-2^{93}$ , então menores valores são melhores:

$$D(y, \theta) = -2 \cdot \underbrace{\sum_{i=1}^N \log \frac{1}{S} \sum_{s=1}^S P(y_i | \theta^s)}_{\text{log pointwise predictive density - lppd}}$$

onde  $N$  é o tamanho da amostra e  $S$  é o número de amostras simuladas da posterior.

<sup>92</sup>divergence

<sup>93</sup>razões históricas

## Interlúdio Histórico – AIC (Akaike, 1973)

$$\text{AIC} = D(y, \theta) + 2k = -2\text{lppd}_{\text{mle}} + 2k$$

onde  $k$  é o número de parâmetros livres do modelo e  $\text{lppd}_{\text{mle}}$  é a estimativa de máxima verossimilhança (MLE) da lppd.

AIC é uma aproximação que somente pode ser confiável quando:

- As *prioris* são uniformes (*flat priors*) ou dominadas totalmente pela função de verossimilhança
- A posterior é aproximadamente uma distribuição Gaussiana/normal multivariada
- O tamanho da amostra  $N$  é muito maior que número de parâmetros livres  $k$ :  $N \gg k$

## Interlúdio Histórico – DIC (Spiegelhalter et al., 2002)

Uma generalização do AIC, onde substituímos a estimação de máxima verossimilhança pela média da posterior e  $k$  por uma correção de viés baseada nos dados:

$$DIC = D(y, \theta) + k_{\text{DIC}} = -2\text{lppd}_{\text{Bayes}} + 2 \left( \underbrace{\text{lppd}_{\text{Bayes}} - \frac{1}{S} \sum_{s=1}^S \log P(y | \theta^s)}_{k \text{ corrigido de viés}} \right)$$

DIC remove a restrição das *prioris* uniformes de AIC, mas mesmo assim mantém os presupostos da posterior ser uma distribuição Gaussiana/normal multivariada e que  $N \gg k$

# Precisão Preditiva

Com o poder computacional que temos hoje não precisamos de aproximações<sup>94</sup>.

Podemos discutir métricas objetivas de **precisão preditiva**.

Mas antes vamos definir o que é precisão preditiva.

---

<sup>94</sup>AIC, DIC etc.

# Precisão Preditiva

## Definição (Precisão Preditiva)

Bayesianos mensuram precisão preditiva usando simulações da distribuição posterior  $\tilde{y}$  do modelo. Para isso temos a distribuição preditiva posterior (*predictive posterior distribution*):

$$p(\tilde{y} \mid y) = \int p(\tilde{y}_i \mid \theta)p(\theta \mid y)d\theta$$

Onde  $p(\theta \mid y)$  é a distribuição posterior do modelo<sup>95</sup>. A fórmula acima significa que calculamos a integral de toda a probabilidade conjunta da distribuição posterior preditiva com a distribuição posterior do nosso modelo. Quanto **maior** a distribuição preditiva posterior  $p(\tilde{y} \mid y)$  **melhor** será a precisão preditiva do modelo.

---

<sup>95</sup>aquela que o rstanarm e brms estima para nós

# Precisão Preditiva

Para mantermos comparabilidade entre amostras, calculamos a esperança dessa medida<sup>96</sup> para cada uma das  $N$  observações da amostra:

$$\text{elpd} = \sum_{i=1}^N \int p_t(\tilde{y}_i) \log p(\tilde{y}_i \mid y) d\tilde{y}$$

onde elpd é esperança do log da densidade preditiva pontual (*expected log pointwise predictive density*) e  $p_t(\tilde{y}_i)$  é a distribuição representando o verdadeiro processo gerativo dos dados para  $\tilde{y}_i$ . Os  $p_t(\tilde{y}_i)$  são desconhecidos e geralmente usamos validação cruzada<sup>97</sup> ou aproximação para a estimativa da elpd.

---

<sup>96</sup>do inglês *expectation* que pode ser também interpretada como a média ponderada

<sup>97</sup>*Cross Validation*

# *Leave-One-Out Cross-Validation (LOO)*

Podemos calcular a elpd usando LOO (Vehtari et al., 2015):

$$\text{elpd}_{\text{loo}} = \sum_{i=1}^N \log p(y_i \mid y_{-i})$$

onde

$$p(y_i \mid y_{-i}) = \int p(y_i \mid \theta) p(\theta \mid y_{-i}) d\theta$$

que é a densidade preditiva com uma observação a menos condicionada nos dados sem a observação  $i$  ( $y_{-i}$ ). Quase sempre usamos a aproximação PSIS-LOO<sup>98</sup> pela sua robustez e baixo custo computacional.

---

<sup>98</sup>mais sobre isso já já...

# *Widely Applicable Information Criteria (WAIC)*

WAIC (Watanabe & Opper, 2010), assim como o LOO também é uma abordagem alternativa para calcularmos a elpd e é definida como:

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{lppd}} - \widehat{p}_{\text{waic}}$$

onde  $\widehat{p}_{\text{waic}}$  é o número estimado efetivo de parâmetros e calculado com base em:

$$\widehat{p}_{\text{waic}} = \sum_{i=1}^N \text{var}_{\text{post}}(\log p(y_i | \theta))$$

que conseguimos calcular usando a variância posterior do log da densidade preditiva para cada observação  $y_i$ :

$$\widehat{p}_{\text{waic}} = \sum_{i=1}^N V_{s=1}^S (\log p(y_i | \theta^s))$$

onde  $V_{s=1}^S$  representa a variância da amostra:

$$V_{s=1}^S a_s = \frac{1}{S-1} \sum_{s=1}^S (a_s - \bar{a})^2$$

## *K-fold Cross-Validation (K-fold CV)*

Da mesma maneira que conseguimos calcular a elpd usando LOO com  $N - 1$  partições da amostra podemos também calcular com qualquer número de partições que quisermos.

Tal abordagem é chamada de **validação cruzada usando  $K$  partições** (*K-fold Cross-Validation*, encurtado para *K-fold CV*).

Ao contrário de LOO, não conseguimos aproximar a elpd usando *K-fold CV* e precisamos fazer a computação atual da elpd sobre  $K$  partições que quase sempre envolve um **alto custo computacional**.

# *Pareto Smoothed Importance Sampling LOO (PSIS-LOO)*

O PSIS usa **amostragem de importância**<sup>99</sup>, o que significa apenas que usa a abordagem de pesos de importância.

A **suavização de Pareto** é uma técnica para tornar os pesos de importância mais confiáveis.

---

<sup>99</sup>*importance sampling*

# Amostragem de Importância (*Importance Sampling*)

Se as  $N$  amostras são condicionalmente independentes<sup>100</sup> (Gelfand et al., 1992) podemos avaliar LOO com amostrad  $\theta^s$  da posterior  $P(\theta | y)$  usando **pesos de importância**:

$$r_i^s = \frac{1}{P(y_i|\theta^s)} \propto \frac{P(\theta^s|y_{-i})}{P(\theta^s|y)}$$

Para então conseguirmos *Importance Sampling Leave-One-Out* (IS-LOO):

$$P(\tilde{y}_i|y_{-i}) \approx \frac{\sum_{s=1}^S r_i^s P(\tilde{y}_i|\theta^s)}{\sum_{s=1}^S r_i^s}$$

---

<sup>100</sup>ou seja são independentes condicionadas aos parâmetros do modelo, que é o pressuposto básico de qualquer modelo probabilístico Bayesiano

# Amostragem de Importância (*Importance Sampling*)

Porém a posterior  $P(\theta | y)$  geralmente possui baixa variância e caudas mais curtas que as distribuições LOO  $P(\theta | y_{-1})$  então se usarmos:

$$P(\tilde{y}_i | y_{-i}) \approx \frac{\sum_{s=1}^S r_i^s P(\tilde{y}_i | \theta^s)}{\sum_{s=1}^S r_i^s}$$

podemos gerar instabilidades pois os  $r_i$  podem ter variância alta ou até infinita.

# Amostragem de Importância com Suavização de Pareto

## *Pareto Smoothed Importance Sampling*

Podemos aprimorar a estimativa IS-LOO usando uma **suavização de Pareto** (*Pareto Smoothed Importance Sampling*) (Vehtari et al., 2015)

Quando a cauda da distribuição dos pesos de importância é longa, um uso direto da amostragem de importância é sensível a um ou alguns valores grandes. Ajustando uma distribuição de Pareto generalizada à cauda superior dos pesos de importância, suavizamos esses valores.

# Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

Por fim temos PSIS-LOO:

$$\widehat{\text{elpd}}_{\text{psis-loo}} = \sum_{i=1}^n \log \left( \frac{\sum_{s=1}^S w_i^s P(y_i | \theta^s)}{\sum_{s=1}^S w_i^s} \right)$$

onde  $w$  é o peso truncado.

# Pareto Smoothed Importance Sampling LOO (PSIS-LOO)

Usamos o parâmetro estimado de forma  $\hat{k}$  da distribuição de Pareto dos pesos de importância para avaliar a confiabilidade da estimativa:

- $k < \frac{1}{2}$  a variância dos pesos de importância é finita, o teorema do limite central se mantém, e a estimativa converge rapidamente
- $\frac{1}{2} < k < 1$  a variância dos pesos de importância é infinita, mas a média existe, o teorema do limite central generalizado para distribuições estáveis se mantém, e a convergência da estimativa é mais lenta. A variação da estimativa PSIS é finita, mas pode ser grande.
- $k > 1$  a variância e a média da distribuição de pesos de importância não existe. A variação da estimativa PSIS é finita, mas pode ser grande

Qualquer valor de  $\hat{k}$  maior que 0.5 é sinal de alerta mas na prática ainda há um bom desempenho com  $\hat{k}$  até 0.7

# Comparação de Modelos no rstanarm

```
library(loo)

loo_1 <- loo(rstanarm_model_1)
loo_2 <- loo(rstanarm_model_2)
loo_3 <- loo(rstanarm_model_3)

loo_compare(loo_1, loo_2, loo_3)
```

# Comparação de Modelos no brms

```
library(loo)

loo_1 <- loo(brms_model_1)
loo_2 <- loo(brms_model_2)
loo_3 <- loo(brms_model_3)

loo_compare(loo_1, loo_2, loo_3)
```

# Referências I

- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. Em B. N. Petrov & F. Csaki (Eds.), *Second International Symposium on Information Theory* (pp. 266–281).
- Amrhein, V., Greenland, S. & McShane, B. (2019). Scientists Rise up against Statistical Significance. *Nature*, 567(7748), 305–307.  
<https://doi.org/10.1038/d41586-019-00857-9>
- Bates, D., Mächler, M., Bolker, B. & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 1–48. <https://doi.org/10.18637/jss.v067.i01>

## Referências II

- Benjamin, D. J., Berger, J. O., Johannesson, M., Nosek, B. A., Wagenmakers, E.-J., Berk, R., Bollen, K. A., Brembs, B., Brown, L., Camerer, C., Cesarini, D., Chambers, C. D., Clyde, M., Cook, T. D., De Boeck, P., Dienes, Z., Dreber, A., Easwaran, K., Efferson, C., ... Johnson, V. E. (2018). Redefine Statistical Significance. *Nature Human Behaviour*, 2(1), 6–10.  
<https://doi.org/10.1038/s41562-017-0189-z>
- Bertsekas, D. P. & Tsitsiklis, J. N. (2008, julho 15). *Introduction to Probability, 2nd Edition* (2nd edition). Athena Scientific.
- Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M. & Stuart, A. (2013). Optimal Tuning of the Hybrid Monte Carlo Algorithm. *Bernoulli*, 19, 1501–1534. <https://doi.org/10.3150/12-BEJ414>

## Referências III

- Betancourt, M. (2017, janeiro 9). *A Conceptual Introduction to Hamiltonian Monte Carlo*. arXiv: 1701.02434. Obtido 6 novembro 2019, de <http://arxiv.org/abs/1701.02434>
- Betancourt, M. (2019, junho). *Probabilistic Building Blocks [Beta & Alpha]*. Obtido 27 maio 2021, de [https://betanalpha.github.io/assets/case\\_studies/probability\\_densities.html](https://betanalpha.github.io/assets/case_studies/probability_densities.html)
- Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM review*, 59(1), 65–98.
- Box, G. E. P. (1976). Science and Statistics. *Journal of the American Statistical Association*, 71(356), 791–799.  
<https://doi.org/10.2307/2286841>
- Brooks, S., Gelman, A., Jones, G. & Meng, X.-L. (2011, maio 10). *Handbook of Markov Chain Monte Carlo*. CRC Press.

## Referências IV

- Bürkner, P.-C. (2017). brms: An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software*, 80(1), 1–28.  
<https://doi.org/10.18637/jss.v080.i01>
- Bürkner, P.-C. (2018). Advanced bayesian multilevel modeling with the r package brms. *The R Journal*, 10(1), 395–411. Obtido 26 fevereiro 2021, de  
<https://journal.r-project.org/archive/2018/RJ-2018-017/index.html>
- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. & Riddell, A. (2017). Stan : A Probabilistic Programming Language. *Journal of Statistical Software*, 76(1). <https://doi.org/10.18637/jss.v076.i01>

## Referências V

- Casella, G. & George, E. (1992). Explaining the Gibbs Sampler [Publisher: Taylor & Francis \_eprint:  
<https://www.tandfonline.com/doi/pdf/10.1080/00031305.1992.10475878>].  
*The American Statistician*, 46(3), 167–174.  
<https://doi.org/10.1080/00031305.1992.10475878>
- Chib, S. & Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4), 327–335.  
<https://doi.org/10.1080/00031305.1995.10476177>
- de Finetti, B. (1974). *Theory of Probability* (Volume 1). John Wiley & Sons.
- Dekking, F. M., Kraaijkamp, C., Lopuhaä, H. P. & Meester, L. E. (2010, outubro 19). *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer.

# Referências VI

- Diaconis, P. & Skyrms, B. (2019, outubro 8). *Ten great ideas about chance* [Google-Books-ID: 68iXDwAAQBAJ]. Princeton University Press.
- Duane, S., Kennedy, A. D., Pendleton, B. J. & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2), 216–222.  
[https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X)
- Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo Method. *Los Alamos Science*, 15(30), 131–136.
- Fisher, R. A. (1925). *Statistical methods for research workers*. Oliver; Boyd.
- Fisher, R. A. (1962). Some Examples of Bayes' Method of the Experimental Determination of Probabilities A Priori. *Journal of the Royal Statistical Society. Series B (Methodological)*, 24(1), 118–124.
- Gabry, J. & Mahr, T. (2021). bayesplot: Plotting for Bayesian Models [R package version 1.8.0]. <https://mc-stan.org/bayesplot/>

## Referências VII

- Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. & Gelman, A. (2019). Visualization in Bayesian Workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2), 389–402.  
<https://doi.org/10.1111/rssc.12378>
- Geisser, S. & Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365), 153–160.
- Gelfand, A. E., Dey, D. K. & Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods. Em J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith (Eds.), *Bayesian Statistics* (pp. 147–167). Oxford University Press.
- Gelfand, A. E. (1996). Model determination using sampling-based methods. *Markov chain Monte Carlo in practice*, 145–161.

## Referências VIII

- Gelman, A. (1992). Iterative and Non-Iterative Simulation Algorithms. *Computing Science and Statistics (Interface Proceedings)*, 24, 457–511.
- Gelman, A. (2008). *The Folk Theorem of Statistical Computing*. [https://statmodeling.stat.columbia.edu/2008/05/13/the\\_folk\\_theore/](https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/)
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. & Rubin, D. B. (2013a). Basics of Markov Chain Simulation. *Bayesian Data Analysis*. Chapman and Hall/CRC.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. & Rubin, D. B. (2013b). *Bayesian Data Analysis*. Chapman and Hall/CRC.
- Gelman, A. & Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge university press.

# Referências IX

- Gelman, A., Hill, J. & Vehtari, A. (2020). *Regression and Other Stories*. Cambridge University Press.
- Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C. & Modrák, M. (2020, novembro 3). *Bayesian Workflow*. arXiv: 2011.01808 [stat]. Obtido 4 fevereiro 2021, de <http://arxiv.org/abs/2011.01808>
- Geman, S. & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(6)*, 721–741.  
<https://doi.org/10.1109/TPAMI.1984.4767596>

# Referências X

- Goodman, S. N. (2016). Aligning Statistical and Scientific Reasoning. *Science*, 352(6290), 1180–1181.  
<https://doi.org/10.1126/science.aaf5406>  
59 citations (Semantic Scholar/DOI) [2021-02-13]
- Goodrich, B., Gabry, J., Ali, I. & Brilleman, S. (2020). rstanarm: Bayesian applied regression modeling via Stan. [R package version 2.21.1].  
<https://mc-stan.org/rstanarm>
- Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1), 97–109.  
<https://doi.org/10.1093/biomet/57.1.97>
- Head, M. L., Holman, L., Lanfear, R., Kahn, A. T. & Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLoS Biol*, 13(3), e1002106.

# Referências XI

- Hoffman, M. D. & Gelman, A. (2011). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1), 1593–1623.  
<http://arxiv.org/abs/1111.4246>
- Ioannidis, J. P. A. (2019). What Have We (Not) Learnt from Millions of Scientific Papers with *P* Values? *The American Statistician*, 73(sup1), 20–25. <https://doi.org/10.1080/00031305.2018.1447512>
- Iserles, A. (2008). *A First Course in the Numerical Analysis of Differential Equations* (2nd). Cambridge University Press.
- It's time to talk about ditching statistical significance. (2019). *Nature*, 567(7748), 283–283. <https://doi.org/10.1038/d41586-019-00874-8>
- Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge university press.

## Referências XII

- Khan, M. E. & Rue, H. (2021, julho 9). *The Bayesian Learning Rule*. arXiv: 2107.04562 [cs, stat]. Obtido 13 julho 2021, de <http://arxiv.org/abs/2107.04562>
- Kolmogorov, A. N. (1933). *Foundations of the Theory of Probability*. Julius Springer.
- Kruschke, J. K. & Vanpaemel, W. (2015). Bayesian Estimation in Hierarchical Models. Em J. R. Busemeyer, Z. Wang, J. T. Townsend & A. Eidels (Eds.), *The Oxford Handbook of Computational and Mathematical Psychology* (pp. 279–299). Oxford University Press Oxford, UK.
- Kurt, W. (2019, julho 9). *Bayesian Statistics the Fun Way: Understanding Statistics and Probability with Star Wars, LEGO, and Rubber Ducks* (Illustrated edition). No Starch Press.

## Referências XIII

- Lakens, D., Adolfi, F. G., Albers, C. J., Anvari, F., Apps, M. A., Argamon, S. E., Baguley, T., Becker, R. B., Benning, S. D., Bradford, D. E., Buchanan, E. M., Caldwell, A. R., Van Calster, B., Carlsson, R., Chen, S. C., Chung, B., Colling, L. J., Collins, G. S., Crook, Z., ... Zwaan, R. A. (2018). Justify Your Alpha. *Nature Human Behaviour*, 2(3), 168–171. <https://doi.org/10.1038/s41562-018-0311-x>
- Lewandowski, D., Kurowicka, D. & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, 100(9), 1989–2001.
- McElreath, R. (2020). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC press.

## Referências XIV

- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6), 1087–1092.  
<https://doi.org/10.1063/1.1699114>
- Muth, C., Oravecz, Z. & Gabry, J. (2018). User-friendly Bayesian regression modeling: A tutorial with rstanarm and shinystan. *Quantitative Methods for Psychology*, 14(2), 99–119.
- Neal, R. M. (1994). An Improved Acceptance Procedure for the Hybrid Monte Carlo Algorithm. *Journal of Computational Physics*, 111(1), 194–203. <https://doi.org/10.1006/jcph.1994.1054>
- Neal, R. M. (2003). Slice Sampling. *The Annals of Statistics*, 31(3), 705–741.

## Referências XV

- Neal, R. M. (2011). MCMC Using Hamiltonian Dynamics. Em S. Brooks, A. Gelman, G. L. Jones & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*.
- Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1), 221–259.
- Neyman, J. (1937). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 236(767), 333–380.
- Perkel, J. M. (2019). Julia: Come for the Syntax, Stay for the Speed. *Nature*, 572(7767), 141–142. <https://doi.org/10.1038/d41586-019-02310-3>

## Referências XVI

- Roberts, G. O., Gelman, A. & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7(1), 110–120.  
<https://doi.org/10.1214/aoap/1034625254>
- Rosnow, R. L. & Rosenthal, R. (1989). Statistical procedures and the justification of knowledge in psychological science. *American Psychologist*, 44, 1276–1284.
- Salvatier, J., Wiecki, T. V. & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2, e55.

## Referências XVII

- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the royal statistical society: Series b (statistical methodology)*, 64(4), 583–639.
- Storopoli, J. (2021). Estatística Bayesiana com R e Stan.  
<https://storopoli.io/Estatistica-Bayesiana>
- van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M. G., Vannucci, M., Gelman, A., Veen, D., Willemsen, J. & Yau, C. (2021). Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1), 1–26.  
<https://doi.org/10.1038/s43586-020-00001-2>
- Van Der Linde, A. (2005). DIC in variable selection. *Statistica Neerlandica*, 59(1), 45–56.

## Referências XVIII

- Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., Bürkner, P.-C., Paananen, T. & Gelman, A. (2020). loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models [R package version 2.4.1].  
<https://mc-stan.org/loo/>
- Vehtari, A., Gelman, A. & Gabry, J. (2015, julho 16). *Practical Bayesian Model Evaluation Using Leave-One-out Cross-Validation and WAIC*. arXiv: 1507.04544. <https://doi.org/10.1101/s11222-016-9696-4>  
1221 citations (Semantic Scholar/DOI) [2021-02-13] 1221 citations (Semantic Scholar/arXiv) [2021-02-13]
- Wasserstein, R. L. & Lazar, N. A. (2016). The ASA's Statement on p-Values: Context, Process, and Purpose. *American Statistician*, 70(2), 129–133. <https://doi.org/10.1080/00031305.2016.1154108>  
2634 citations (Semantic Scholar/DOI) [2021-02-13]

# Referências XIX

- Wasserstein, R. L., Schirm, A. L. & Lazar, N. A. (2019). Moving to a World Beyond “ $p < 0.05$ ”. *American Statistician*, 73, 1–19.  
<https://doi.org/10.1080/00031305.2019.1583913>  
662 citations (Semantic Scholar/DOI) [2021-02-13]
- Watanabe, S. & Opper, M. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory.. *Journal of machine learning research*, 11(12).

# Licença

O texto e as figuras desses slides possuem uma  
Licença Creative Commons  
Atribuição-NãoComercial-Compartilhamento 4.0  
Internacional (CC BY-NC-SA 4.0)



## Como citar esse Conteúdo

Storopoli, J. (2021). Estatística Bayesiana com R e Stan.  
<https://storopoli.io/Estatistica-Bayesiana>

# Simulação Monte Carlo do Problema de Monty Hall em R

```
monty <- function(){
  door_car <- sample(1:3L, 1)
  door_pick <- sample(1:3L, 1)
  door_goats <- setdiff(1:3L, door_car)
  if(door_pick == door_car) {
    door_show <- sample(door_goats, 1)
  } else{
    door_show <- setdiff(door_goats, door_pick)
  }
  door_switch <- setdiff(1:3L, c(door_pick, door_show))[1]
  door_pick <- door_switch
}
wins <- 0
ntrials <- 10000
for (i in 1:ntrials){
  wins <- wins + monty()
}
wins / ntrials
```

# Simulação Monte Carlo do Problema de Monty Hall em Julia

```
function monty()
    door_car = rand(1:3)
    door_pick = rand(1:3)
    door_goats = setdiff([1;2;3],door_car)
    door_show = door_pick == door_car ? rand(door_goats) : setdiff(door_goats,door_pick)
    door_switch = setdiff([1;2;3], [door_pick; door_show])[1]
    door_pick = door_switch
    return door_pick == door_car
end
ntrials = 10_000
wins = reduce(+, (monty() for i ∈ 1:ntrials))
wins / ntrials
```

# Como surgiu a distribuição Normal<sup>101102</sup>

$$\text{Binomial}(n, k) = \binom{n}{k} p^k (1-p)^{n-k}$$
$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \quad (\text{Stirling})$$

$$\lim_{n \rightarrow \infty} \binom{n}{k} p^k (1-p)^{n-k} = \frac{1}{\sqrt{2\pi npq}} e^{-\frac{(k-np)^2}{2npq}}$$

Sabemos que na binomial:  $E = np$  e  $\text{Var} = npq$ ; logo substituindo E por  $\mu$  Var por  $\sigma^2$ :

$$\lim_{n \rightarrow \infty} \binom{n}{k} p^k (1-p)^{n-k} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(k-\mu)^2}{\sigma^2}}$$

---

<sup>101</sup>Abraham de Moivre em 1738

<sup>102</sup>Uma melhor explanação pode ser encontrada clicando [aqui](#)

# Buscas Scopus do Stan

- Uso Geral na Scopus: ALL((brms AND burkner) OR (gelman AND hoffman AND stan) OR mc-stan.org OR rstanarm OR pystan OR (rstan AND NOT mit))
- Uso em Comparaçao com outras Ferramentas Bayesianas:  
REF((gelman AND hoffman AND stan) OR mc-stan.org)  
AND REF(brms AND burkner) AND REF(pystan) AND  
REF(rstanarm) AND REF(rstan AND NOT mit)
- Uso das Ferramentas Bayesianas: a busca anterior adicionado de  
REF(PyMC3 OR (PyMC\* AND fonnesbeck)) AND  
REF(tensorflow) AND REF(pytorch) AND REF(Keras)

# Decomposição QR

Em Álgebra Linear 101 aprendemos que qualquer matriz (até mesmo as retangulares) podem ser decompostas em um produto de duas matrizes:

- $\mathbf{Q}$ : uma matriz ortogonal (suas colunas são vetores unitários ortogonais, i.e.  $\mathbf{Q}^T = \mathbf{Q}^{-1}$ )
- $\mathbf{R}$ : uma matrix triangular superior

Agora vamos incorporar a decomposição QR no modelo de regressão linear. Aqui, usarei o QR "fino" em vez do "gordo", que escala  $\mathbf{Q}$  e  $\mathbf{R}$  matrizes por um fator de  $\sqrt{n - 1}$  onde  $n$  é o número de linhas de  $\mathbf{X}$ . Na prática, é melhor implementar a decomposição QR fina, que é preferível à decomposição QR gorda. É numericamente mais estável. Matematicamente, a decomposição QR fina é:

$$\mathbf{X} = \mathbf{Q}^* \mathbf{R}^*$$

$$\mathbf{Q}^* = \mathbf{Q} \cdot \sqrt{n - 1}$$

$$\mathbf{R}^* = \frac{1}{\sqrt{n - 1}} \cdot \mathbf{R}$$

$$\begin{aligned}\boldsymbol{\mu} &= \alpha + \mathbf{X} \cdot \boldsymbol{\beta} + \sigma \\ &= \alpha + \mathbf{Q}^* \cdot \mathbf{R}^* \cdot \boldsymbol{\beta} + \sigma \\ &= \alpha + \mathbf{Q}^* \cdot (\mathbf{R}^* \cdot \boldsymbol{\beta}) + \sigma \\ &= \alpha + \mathbf{Q}^* \cdot \tilde{\boldsymbol{\beta}} + \sigma\end{aligned}$$