

UNIVERSIDADE DE SÃO PAULO

Programa de Pos-Graduação em Estatística

Alex Monito Nhancololo

1 Questão 1:

Seja $(\mathcal{X}, \mathcal{A}, \mathcal{P})$ um modelo estatístico, onde $\mathcal{X} \subseteq \mathbb{R}^n$ é o espaço amostral (o conjunto de todos os resultados $X = (x_1, \dots, x_n)$ do experimento) e $n \in \mathbb{N}$ é o tamanho da amostra. O componente \mathcal{A} é uma σ -álgebra sobre \mathcal{X} (tipicamente a σ -álgebra de Borel) que define o conjunto de todos os eventos de interesse. A família $\mathcal{P} = \{P_\theta, \theta \in \Theta \subseteq \mathbb{R}^k\}$ é o conjunto de medidas de probabilidade definidas sobre o espaço mensurável $(\mathcal{X}, \mathcal{A})$, representando todas as explicações probabilísticas consideradas para os dados. Neste modelo, θ é o vetor de parâmetros (desconhecido) que indexa as distribuições, Θ é o espaço paramétrico (o conjunto de todos os valores possíveis para θ), e $k \in \mathbb{N}$ é a dimensão do espaço paramétrico [Patriota, 2013, 2014, 2017].

Uma **Hipótese Geral** é formalmente definida como uma afirmação que restringe o parâmetro a um subconjunto Θ_0 do espaço total, ou seja, $H_0 : \theta \in \Theta_0$, onde $\Theta_0 \subseteq \Theta$ [Patriota, 2013]. Esta notação unifica os diferentes tipos de testes de hipóteses, que são apenas casos especiais de Θ_0 : como à unilateral a esquerda ($\Theta_0 = \{\theta : \theta \leq \theta_0\}$, isto é, $H_0 : \theta \leq \theta_0$), unilateral à direita ($\Theta_0 = \{\theta : \theta \geq \theta_0\}$, isto é, $H_0 : \theta \geq \theta_0$), bilateral ($\Theta_0 = \{\theta : \theta = \theta_0\}$, isto é, $H_0 : \theta = \theta_0$) ou intervalares ($\Theta_0 = \{\theta : \theta \in [\theta_i, \theta_j]\}$), definidas em Schervish [1996], onde $\theta_0, \theta_i, \theta_j, i \neq j$ são constantes pré-definidas pelo pesquisador. Para testar H_0 calcula-se uma estatística de teste $T_{H_0}(X)$ (que resume os dados, ex: diferença entre médias) e seu valor observado (obtida substituindo os dados observados) $t_{obs} = T_{H_0}(x)$.

O **valor-p**, $p(H_0, t)$, é então definido como a probabilidade, no pior caso, de se observar uma estatística de teste $T_{H_0}(X)$ pelo menos tão extrema quanto t_{obs} , calculada sobre todos os parâmetros θ sob H_0 [Patriota, 2014, Wasserstein and Lazar, 2016]:

$$p(H_0, t) = \sup_{\theta \in \Theta_0} P_\theta(T_{H_0}(X) \geq t_{obs}).$$

Se o objetivo fosse testar especificamente se a medida geradora dos dados é P_{θ_i} ou $P_{\theta_{i'}}$ (com $i \neq i'$), a hipótese nula seria $H_0 : P_\theta \in \{P_{\theta_i}, P_{\theta_{i'}}\}$. Neste caso, como Θ_0 contém apenas dois elementos, o supremo (sup) torna-se simplesmente o máximo (max) entre as duas probabilidades [Patriota, 2014]:

$$p(H_0, t) = \max\{P_{\theta_i}(T_{H_0}(X) \geq t), P_{\theta_{i'}}(T_{H_0}(X) \geq t)\}.$$

Valor-p Aproximado $p_a(H_0, t)$

O valor-p aproximado (ou assintótico) é usada para testar hipóteses gerais ($H_0 : \theta \in \Theta_0$), especialmente quando o cálculo do valor-p $\{p(H_0, t)\}$ exato (baseado no sup) é complexo ou impossível. Vários testes podem ser usados para obter uma estatística de teste $T_{\Theta_0}(X)$, cujo valor observado t_{obs} será usado no cálculo do p_a . As três abordagens principais, que são assintoticamente equivalentes sob H_0 , são:

1. Teste da Razão de Verossimilhanças (TRV): Adequado para hipóteses gerais, não apenas simples.

$$\bullet H_0 : \theta \in \Theta_0, \quad \lambda_{\Theta_0}(x) = \frac{\sup_{\theta \in \Theta_0} L(\theta, x)}{\sup_{\theta \in \Theta} L(\theta, x)}, \quad T_{RV_{\Theta_0}}(X) = -2 \log(\lambda_{\Theta_0}(X)) = 2(l(\hat{\theta}) - l(\hat{\theta}_0))$$

2. Teste de Wald: Particularmente útil para hipóteses lineares.

$$\bullet H_0 : C_{q \times k} \cdot \theta = d_{q \times 1}, \quad T_{W_{\Theta_0}}(X) = (C_{q \times k} \hat{\theta} - d_{q \times 1})^\top [C \hat{A} C_{q \times 1}^\top]^{-1} (C \hat{\theta} - d) \text{ (onde } \hat{A} \text{ é a matriz de covariância estimada de } \hat{\theta})$$

3. Teste Escore (Teste multiplicador de Lagrange -LM):

$$\bullet H_0 : \theta \in \Theta_0, \quad (\text{Vetor Escore}) U(\theta) = \frac{\partial l(\theta, x)}{\partial \theta}, \quad (\text{Matriz de Informação}) I(\theta) = -E \left[\frac{\partial^2 l(\theta, X)}{\partial \theta \partial \theta^\top} \right], \\ T_{S_{\Theta_0}}(X) = U(\hat{\theta}_0)^\top [I(\hat{\theta}_0)]^{-1} U(\hat{\theta}_0) \text{ (calculada na estimativa restrita } \hat{\theta}_0)$$

Conforme descrito em Patriota [2013, 2017], sob H_0 e condições de regularidade, a estatística de teste para qualquer uma dessas abordagens ($T_{\text{Teste}_{\Theta_0}}(X)$, onde $\text{Teste} \in \{LR, W, S\}$) converge em distribuição para uma Qui-Quadrado com q graus de liberdade ($T_{\text{Teste}_{\Theta_0}}(X) \xrightarrow{d} \chi_q^2$), onde $q = \dim(\Theta) - \dim(\Theta_0)$, em que $\dim(\Theta)$ é o número de parâmetros

distintos que precisam ser estimados no modelo completo (modelo sem restrições de H_0) e $\dim(\Theta_0)$ é o número de parâmetros livres/ que sobram para estimar após aplicar essas restrições (H_0). O valor-p aproximado (p_a) é então calculado como a probabilidade de cauda da estatística observada (t_{obs}) em relação a esta distribuição χ_q^2 :

$$p_a(\Theta_0; x) = P(\chi_q^2 \geq t_{obs}) = 1 - F_{\chi_q^2}(t_{obs})$$

onde t_{obs} é o valor observado da estatística $T_{\text{Teste}_{\Theta_0}}(X)$ e $F_{\chi_q^2}$ é a função distribuição acumulada da χ_q^2 .

Discussão dos Problemas da Interpretação "Condicional" ($P(T_{\Theta_0}(X) > t_{obs} | H_0)$)

É comum encontrar o valor-p definido usando a notação informal $P(T_{\Theta_0}(X) > t_{obs} | H_0)$. No entanto, esta é informal e "inadequada" pois mascara a verdadeira definição (o supremo) e sugere propriedades que o valor-p não possui:

- No modelo clássico, $H_0 : \theta \in \Theta$ e $T_{\Theta_0}(X) > t_{obs}$ não pertencem ao mesmo espaço de probabilidade, tornando o condicionamento em H_0 um abuso de notação. Isto é, $P(T_{\Theta_0}(X) > t_{obs} | H_0) = P(T_{\Theta_0}(X) > t_{obs} \cap H_0) / P(H_0)$ exige que $\{T_{\Theta_0}(X) > t_{obs}\} \in \mathcal{X}$ e $H_0 \in \Theta$ sejam eventos no mesmo espaço de probabilidade (na mesma σ -álgebra), o que não ocorre. H_0 não é um evento aleatório, é uma afirmação sobre qual medida de probabilidade (indexada por $\theta \in \Theta_0$) governa os dados, enquanto $T_{\Theta_0}(X)$ é evento aleatório, dependendo de X a estimativa muda. Como Wasserstein and Lazar [2016] adverte, o valor-p não mede a probabilidade de H_0 ser verdadeira.
- Incoerência lógica. Se $H_1 : \theta \in \Theta_1$ implica $H_2 : \theta \in \Theta_2$ (ou seja, $\Theta_1 \subseteq \Theta_2$), a evidência contra H_1 (a hipótese mais restrita) deveria ser pelo menos tão forte quanto a evidência contra H_2 . O valor-p viola esta regra, levando a conclusões invertidas onde $p_2 < p_1$ [Schervish, 1996, Patriota, 2013, 2014]. Patriota [2013] apresenta um caso onde $H_1 : \mu_1 = \mu_2 = 2$ (contida em $H_2 : \mu_1 = \mu_2$) resultou em $p_1(H_1, t) = 0.10$ e $p_2(H_2, t) = 0.03$. Isso leva à conclusão de rejeitar $\mu_1 = \mu_2$, mas não rejeitar $\mu_1 = \mu_2 = 2$ 99. Schervish [1996] mostra um caso com $H_1 : \theta \in [-0.5, 0.5]$ (contida em $H_2 : \theta \in [-0.82, 0.52]$). Os dados geraram $p(H_1, t) = 0.0502$ e $p(H_2, t) = 0.0498$, que também rejeita-se a hipótese maior (H_2) mas não a hipótese menor (H_1) contida nela, o que é absurdo. A causa desta incoerência é que a métrica (a distribuição de referência χ_q^2) usada para calcular o valor-p (aproximado) muda para cada hipótese, pois q (os graus de liberdade) depende de $\dim(\Theta_0)$. Como a métrica muda, o suporte medido pelo valor-p depende da hipótese. Isso significa que um $p = 0.05$ para um teste de $H_0 : \theta = \theta_0$ não é comparável a um $p = 0.05$ para um teste unilateral ($H_0 : \theta \leq \theta_0$).
- Um valor-p muito pequeno (ex: $p = 0.001$) não significa que o efeito (ex: a diferença entre médias ou o coeficiente β_1) é grande ou importante. O valor-p é uma função tanto do tamanho do efeito como do tamanho da amostra (n). Com amostras muito grandes, efeitos triviais e sem importância prática podem gerar valores-p muito pequenos.
- Um valor-p não significativo (ex: $p = 0.50$) não prova que H_0 é verdadeira ou que o efeito é zero. Significa apenas que os dados não são suficientemente incompatíveis com H_0 .

Exemplo: Seja $Y_i | X_i = x_i \sim \text{Poisson}(\lambda_i)$ a variável resposta (ex: número de nascidos vivos na unidade i) e X_i a variável explicativa (ex: experiência em anos do médico i responsável pelo parto), $i = 1, \dots, n$. Supondo que desejamos modelar a média λ_i como função de X_i . Como a média da Poisson deve ser positiva ($\lambda_i > 0$), usamos a função de ligação logarítmica (que é a canônica) para mapear o preditor linear $\eta_i \in (-\infty, \infty)$ para o domínio $(0, \infty)$:

$$g(\lambda_i) = \log(\lambda_i) = \eta_i = \beta_0 + \beta_1 X_i$$

O modelo para a média é, portanto, $\lambda_i = \exp(\beta_0 + \beta_1 X_i)$, o que garante $\lambda_i > 0$ e o vetor de parâmetros é $\theta = (\beta_0, \beta_1)^\top$. Suponhamos que o objetivo seja testar se a experiência em anos (X_i), do médico tem efeito sobre o número de nascidos vivos (Y_i). A hipótese fica, $H_0 : \beta_1 = 0$ vs $H_1 : \beta_1 \neq 0$, e, o espaço paramétrico irrestrito (sem considerar H_0) é $\Theta = \{(\beta_0, \beta_1) \in \mathbb{R}^2\}$ (que tem dois parâmetros e ambos precisam ser estimados, portanto $\dim(\Theta) = 2$), e o espaço paramétrico restrito (sob H_0) é $\Theta_0 = \{(\beta_0, \beta_1) \in \mathbb{R}^2 \mid \beta_1 = 0\}$ (tem dois parâmetros mas um é fixo (β_1) e apenas β_0 é que precisa ser estimado, portanto $\dim(\Theta_0) = 1$). Assim, os graus de liberdade para o teste assintótico são $q = \dim(\Theta) - \dim(\Theta_0) = 2 - 1 = 1$.

Função de Log-Verossimilhança

$$P(Y_i = y_i) = \frac{e^{-\lambda_i} \lambda_i^{y_i}}{y_i!} = \frac{e^{-\exp\{\beta_0 + \beta_1 x_i\}} [\exp\{\beta_0 + \beta_1 x_i\}]^{y_i}}{y_i!}$$

$$L(\beta_0, \beta_1) = \prod_{i=1}^n P(Y = y_i) = \prod_{i=1}^n \frac{e^{-\exp(\beta_0 + \beta_1 x_i)} \exp(y_i[\beta_0 + \beta_1 x_i])}{y_i!}$$

$$\log L(\beta_0, \beta_1) = \sum_{i=1}^n [-\exp(\beta_0 + \beta_1 x_i) + y_i(\beta_0 + \beta_1 x_i) - \log(y_i!)]$$

Modelo irrestrito

$$\begin{cases} \frac{\partial \log L(\beta_0, \beta_1)}{\partial \beta_0} = \sum_{i=1}^n [y_i - \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i)] = 0 \\ \frac{\partial \log L(\beta_0, \beta_1)}{\partial \beta_1} = \sum_{i=1}^n [y_i x_i - x_i \exp(\hat{\beta}_0 + \hat{\beta}_1 x_i)] = 0 \end{cases} \quad (\hat{\beta}_0 \text{ e } \hat{\beta}_1 \text{ não tem uma forma bonita})$$

$$\text{Modelo restrito a } H_0 : \beta_1 = 0 \implies \log L(\beta_0, \beta_1) = \sum_{i=1}^n [-\exp(\beta_0) + y_i(\beta_0) - \log(y_i!)]$$

$$\frac{\partial \log L(\beta_0)}{\partial \beta_1} = \sum_{i=1}^n [y_i - \exp(\hat{\beta}_0)] = 0 \implies \mathcal{N} \bar{y} = \mathcal{N} \exp(\hat{\beta}_0), \implies \hat{\beta}_{0_{H_0}} = \log(\bar{y}), \bar{y} \neq 0$$

Estatística TRV

$$T_{RV_{\Theta_0}}(X) = 2[\log L(\hat{\beta}_0, \hat{\beta}_1) - \log L(\hat{\beta}_{0_{H_0}}, \hat{\beta}_{1_{H_0}} = 0)]$$

$$= 2 \left[\sum_{i=1}^n [-\exp(\hat{\beta}_0 + \hat{\beta}_1 x_i) + y_i(\hat{\beta}_0 + \hat{\beta}_1 x_i) - \log(y_i!)] - \sum_{i=1}^n [-\exp(\log(\bar{y})) + y_i(\log(\bar{y})) - \log(y_i!)] \right]$$

$$= 2 \left[\sum_{i=1}^n [-\exp(\hat{\beta}_0 + \hat{\beta}_1 x_i) + y_i(\hat{\beta}_0 + \hat{\beta}_1 x_i)] - \sum_{i=1}^n [-\bar{y} + y_i(\log(\bar{y}))] \right]$$

Daqui precisaremos gerar os dados e obter tudo numericamente

$$p_a = P(\chi_1^2 \geq T_{RV_{\Theta_0}}(X)) = 1 - F_{\chi_1^2}(T_{RV_{\Theta_0}}(X))$$

Figure 1: A) Distribuição do número de nascidos vivos e, B) modelo ajustado

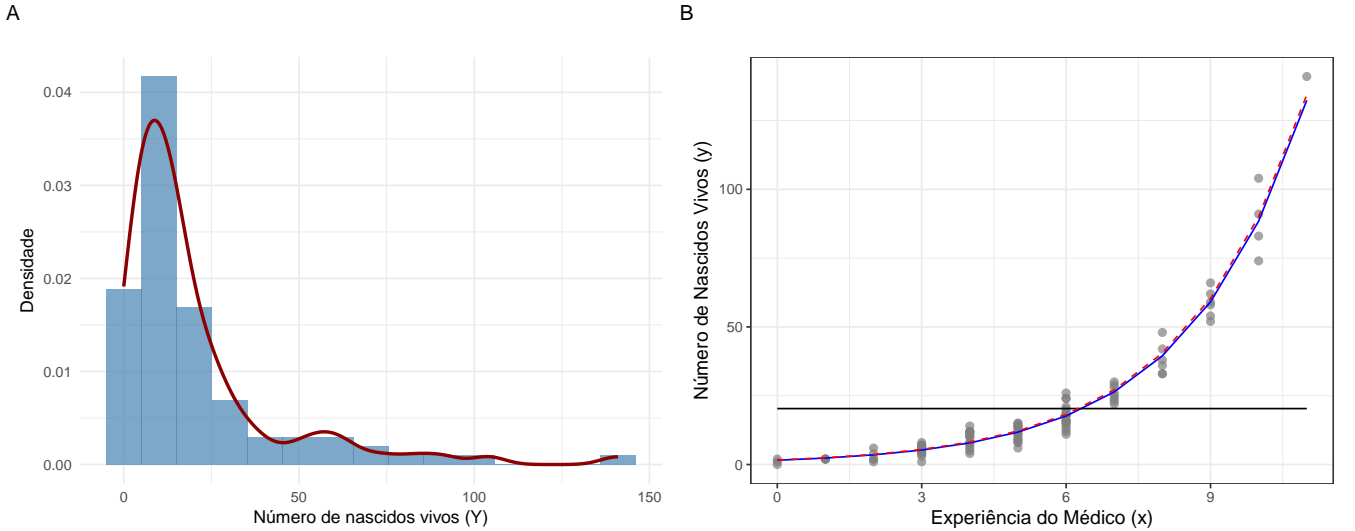


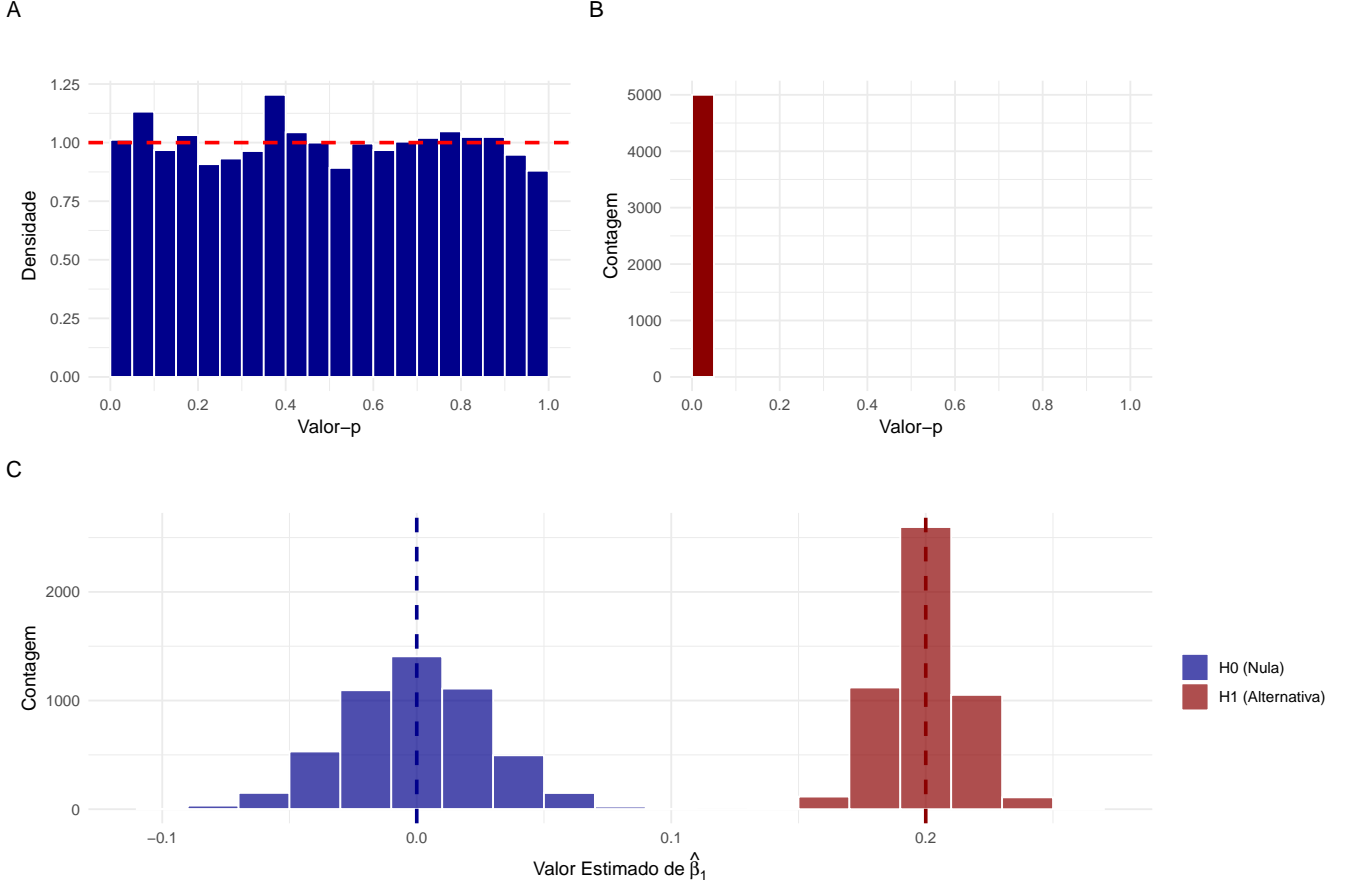
Table 1: Resultados do teste da razão de verossimilhanças (TRV) e estimativas dos parâmetros

Modelo	β_0	β_1	Log-Verossimilhança	Estatística TRV (t_{obs})	Valor-p Aproximado (p_a)
Completo	0.4504	0.4032	-256.0587	1887.8952	<
Restrito (Nulo)	3.0121	0	-1200.0063		0.000001

Simulações Monte Carlo

A Figura 2 mostra o desempenho do teste de razão de verossimilhança. O Gráfico A mostra que o teste é bem cali-

Figure 2: Avaliação do desempenho do teste de razão de verossimilhança (LRT) por simulação de Monte Carlo (5000 simulações) para regressão Poisson. A) Distribuição dos valores-p sob H_0 , demonstrando o controle do Erro Tipo I, B) Distribuição dos valores-p sob H_1 , demonstrando o poder do teste, C) Distribuição das estimativas $\hat{\beta}_1$ em ambos os cenários.



brado, pois, quando a hipótese nula H_0 é verdadeira (efeito $\beta_1 = 0$), os p-valores devem seguir distribuição Uniforme, controlando corretamente o Erro Tipo I. O Gráfico B demonstra o poder do teste: quando a hipótese alternativa H_1 é verdadeira (efeito $\beta_1 = 0.2$), os p-valores se concentram massivamente perto de zero, indicando que o teste detecta o efeito real com sucesso. O Gráfico C mostra que as distribuições das estimativas de $\hat{\beta}_1$ são claramente separadas. Sob H_0 (azul), elas se centram em 0, enquanto sob H_1 (vermelho), se centram em 0.2, permitindo ao teste distinguir o ruído aleatório de um efeito verdadeiro.

2 Questão 2

O Teorema da Aproximação Universal, inicialmente provado por Funahashi [1989] estabelece uma condição **suficiente** para que redes neurais *feedforward* de três camadas (uma camada oculta) pode aproximar qualquer função contínua $f : K \rightarrow \mathbb{R}$, definida em um subconjunto compacto $K \subset \mathbb{R}^n$, com qualquer grau de precisão $\epsilon > 0$. Isto é, Seja $K \subset \mathbb{R}^n$ um conjunto compacto e $f : K \rightarrow \mathbb{R}$ uma função contínua. Se a função de ativação ϕ , $\phi : \mathbb{R} \rightarrow \mathbb{R}$ é **não constante, limitada, monotônica crescente e contínua**, então para qualquer grau de precisão $\epsilon > 0$, existem N neurônios na camada oculta $N \in \mathbb{N}$, $w_{ij} (i = 1, \dots, N, j = 1, \dots, n)$ pesos da camada de entrada para a camada oculta $w_{ij} \in \mathbb{R}$, b_i vieses (ou limiares) $b_i \in \mathbb{R}$, e c_i pesos da camada oculta para a saída $c_i \in \mathbb{R}$ tais que:

$$\max_{\mathbf{x} \in K} \left| f(x_1, \dots, x_n) - \tilde{f}(x_1, \dots, x_n) \right| < \epsilon, \quad \tilde{f}(x_1, \dots, x_n) = \sum_{i=1}^N c_i \phi \left(\sum_{j=1}^n w_{ij} x_j + b_i \right), \quad \forall \mathbf{x} = (x_1, \dots, x_n) \in K,$$

onde $\tilde{f}(\mathbf{x})$ é a função de aproximação.

Generalização do teorema de Funahashi

Uma rede neural $\tilde{f}(\mathbf{x})$, com uma função de ativação ϕ que seja localmente limitada e contínua por partes, pode aproximar qualquer função contínua $f(\mathbf{x}) \in C(R^n)^1$ no conjunto compacto $K \subset R^n$ se, e somente se, a função de ativação ϕ não for um polinômio [Leshno et al., 1993].

A generalização de Leshno et al. [1993] relaxa os requisitos de Funahashi [1989] de três maneiras:

- **Limitação:** Funahashi [1989] exige que ϕ seja limitada enquanto Leshno et al. [1993] permite funções não limitadas, como a popular ReLU, desde que sejam localmente limitadas.
- **Continuidade:** Funahashi [1989] exige que ϕ seja contínua, enquanto Leshno et al. [1993] permite funções descontínuas (contínuas por partes), como a função de grau enquanto
- **Monotonicidade:** Funahashi [1989] exige que ϕ seja monotônica, enquanto Leshno et al. [1993] remove essa exigência, permitindo funções oscilatórias.

⇒ **Exemplos:**

1. Rede com ativação ReLU ($\phi(x) = \max(0, x)$): O teorema de Funahashi [1989] não se aplica, pois a ReLU não é limitada. No entanto, o teorema de Leshno et al. [1993] garante que a rede ReLU é um aproximador universal, pois $\max(0, x)$ não é um polinômio.
2. Rede com ativação Polinomial ($\phi(x) = x^2$): O teorema de Leshno et al. [1993] prova que esta rede falhará em ser um aproximador universal. A rede só pode gerar funções da forma $\tilde{f}(x) = \sum c_i(w_i \cdot x + b_i)^2$, que é meramente um polinômio de grau 2. Tal rede nunca poderia aproximar arbitrariamente bem uma função não polinomial, como $\sin(x)$ ou e^x .
3. Rede com $\phi(x) = \sin(x)$: O teorema de Funahashi [1989] não se aplica (pois $\sin(x)$ não é monotônica). O teorema de Leshno et al. [1993] se aplica (pois $\sin(x)$ não é um polinômio), mas também ilustra a importância do limiar b_i . Uma rede sem limiar, $\sum c_i \sin(w_i \cdot x)$, só pode criar funções ímpares e falharia em aproximar uma função par como $\cos(x)$. O limiar b_i é essencial, pois permite o deslocamento de fase, $\sin(w \cdot x + b_i)$, que é necessário para a universalidade.

3 Questão 3

Seja $Y_i|X = x_i \sim \text{Binomial}(m_i, \mu_i)$ para $i = 1, \dots, n$, onde Y_i é o número de "sucessos", m_i é o número total de tentativas (conhecido) e $\mu_i \in [0, 1]$ é a probabilidade de sucesso. O vetor de parâmetros $\theta_{p \times 1} \in \mathbb{R}^p$ é usado para modelar μ_i através de um preditor linear $\eta_i = x_i^\top \theta = x_{i1}\theta_1 + x_{i2}\theta_2 + \dots + x_{ip}\theta_p = \sum_{j=1}^p x_{ij}\theta_j$ (a j -ésima covariável $\{x_i\}_{p \times 1}$ para a i -ésima observação) e uma função de ligação $g(\cdot)$. Usamos a ligação canônica logística (logit):

$$g(\mu_i) = \log\left(\frac{\mu_i}{1 - \mu_i}\right) = \eta_i = x_i^\top \theta, \quad \mu_i = g^{-1}(\eta_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} = \frac{1}{1 + \exp(-\eta_i)}, \quad P(Y_i = y_i) = \binom{m_i}{y_i} \mu_i^{y_i} (1 - \mu_i)^{m_i - y_i}.$$

Função de Log-Verossimilhança

$$\begin{aligned} L(\theta) &= \prod_{i=1}^n P(Y_i = y_i) = \prod_{i=1}^n \binom{m_i}{y_i} \mu_i^{y_i} (1 - \mu_i)^{m_i - y_i} \\ \ell(\theta) &= \log L(\theta) = \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i \log(\mu_i) + (m_i - y_i) \log(1 - \mu_i) \right] \\ &= \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i \log\left(\frac{\exp(\eta_i)}{1 + \exp(\eta_i)}\right) + (m_i - y_i) \log\left(1 - \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}\right) \right] \\ &= \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i [\log(\exp(\eta_i)) - \log(1 + \exp(\eta_i))] + (m_i - y_i) \log\left(\frac{1 + \exp(\eta_i) - \exp(\eta_i)}{1 + \exp(\eta_i)}\right) \right] \\ &= \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i [\eta_i - \log(1 + \exp(\eta_i))] + (m_i - y_i) \log\left(\frac{1}{1 + \exp(\eta_i)}\right) \right] \\ &= \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i \eta_i - y_i \log(1 + \exp(\eta_i)) - (m_i - y_i) \log(1 + \exp(\eta_i)) \right] \\ &= \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i \eta_i - [y_i + m_i - y_i] \log(1 + \exp(\eta_i)) \right] \\ \ell(\theta) &= \sum_{i=1}^n \left[\log \binom{m_i}{y_i} + y_i \eta_i - m_i \log(1 + \exp(\eta_i)) \right] \end{aligned}$$

¹Pela teoria da análise funcional, equivale a dizer que $\tilde{f}(\mathbf{x})$ é densa em $C(R^n)$

Função Escore (Estimação)

Pela teoria de funções de estimação (FE), a FE da MV é a função escore [Desmond and Godambe, 2005], $U(\theta)$, cujas componentes $j = 1, \dots, p$ são as derivadas parciais de $\ell(\theta)$ em relação a cada θ_j . Usamos a regra da cadeia:

$$\begin{aligned}
 U_j(\theta) &= \frac{\partial \ell(\theta)}{\partial \theta_j} = \sum_{i=1}^n \frac{\partial \ell_i(\theta)}{\partial \theta_j} = \sum_{i=1}^n \frac{\partial \ell_i(\theta)}{\partial \eta_i} \cdot \frac{\partial \eta_i}{\partial \theta_j} \\
 \frac{\partial \eta_i}{\partial \theta_j} &= \frac{\partial (x_i^\top \theta)}{\partial \theta_j} = \frac{\partial (\sum_{j=1}^p x_{ij} \theta_j)}{\partial \theta_j} = x_{ij} \\
 \frac{\partial \ell_i(\theta)}{\partial \eta_i} &= \frac{\partial}{\partial \eta_i} \left[\log \left(\frac{m_i}{y_i} \right) + y_i \eta_i - m_i \log(1 + \exp(\eta_i)) \right] \\
 \frac{\partial \ell_i(\theta)}{\partial \eta_i} &= 0 + y_i - m_i \cdot \left(\frac{1}{1 + \exp(\eta_i)} \right) \cdot \frac{\partial (1 + \exp(\eta_i))}{\partial \eta_i} = y_i - m_i \left(\frac{1}{1 + \exp(\eta_i)} \right) \cdot \exp(\eta_i) = y_i - m_i \left(\frac{\exp(\eta_i)}{1 + \exp(\eta_i)} \right) \\
 \frac{\partial \ell_i(\theta)}{\partial \eta_i} &= y_i - m_i \mu_i \quad (\text{pois } \mu_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}) \\
 U_j(\theta) &= \frac{\partial \ell(\theta)}{\partial \theta_j} = \sum_{i=1}^n \frac{\partial \ell_i(\theta)}{\partial \theta_j} = \sum_{i=1}^n \frac{\partial \ell_i(\theta)}{\partial \eta_i} \cdot \frac{\partial \eta_i}{\partial \theta_j} = \sum_{i=1}^n (y_i - m_i \mu_i) x_{ij} \\
 U(\theta) &= \begin{bmatrix} \frac{\partial \ell(\theta)}{\partial \theta_1} \\ \frac{\partial \ell(\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial \ell(\theta)}{\partial \theta_p} \end{bmatrix} = \begin{bmatrix} U_{j=1}(\theta) \\ U_2(\theta) \\ \vdots \\ U_p(\theta) \end{bmatrix} \\
 U(\theta) &= \sum_{i=1}^n (Y_i - m_i \mu_i) x_i, \quad x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^\top \\
 \mathbb{E}[U(\theta)] &= \mathbb{E} \left[\sum_{i=1}^n (Y_i - m_i \mu_i) x_i \right] = \sum_{i=1}^n x_i (\mathbb{E}[Y_i] - m_i \mu_i) = 0, \quad \mathbb{E}[Y_i] = m_i \mu_i, \quad \text{logo } U(\theta) \text{ é FE não viesada}
 \end{aligned}$$

3 Função de Estimação de robusta

O modelo binomial pertence à família exponencial, e, pela teoria de GLM, sabemos que $V(Y_i) = \phi \mathbb{E}[Y_i](1 - \mu_i) = \phi m_i \mu_i (1 - \mu_i)$, onde $\phi > 0$ é o parâmetro de dispersão e, para $\phi = 1$, temos o modelo binomial [Wedderburn, 1974]. Pela teoria das FE, uma FE $U(\theta)$ deve ser não-viesada, satisfazendo $\mathbb{E}[U(\theta)] = 0$ para todo θ . Para comparar a eficiência de FEs $U(\theta)$ não-viesadas dentro de uma classe \mathcal{G} , usa-se a versão padronizada $U^*(\theta) = U(\theta)/\mathbb{E}[\partial U(\theta)/\partial \theta]$. A FE $U_Q(\theta)$ é dita ótima em \mathcal{G} se sua versão padronizada $U_Q^*(\theta)$ satisfaz $\mathbb{E}[(U_Q^*(\theta))^2] \leq \mathbb{E}[(U^*(\theta))^2]$ para toda outra $U(\theta) \in \mathcal{G}$. Especificamente, para um modelo semiparamétrico onde $E(Y_i) = \alpha_i(\theta)$ e $\text{var}(Y_i) = \sigma_i^2(\theta)$, a FE ótima $U_Q(\theta)$ dentro da classe \mathcal{G}_l de combinações lineares das FEs elementares $h_i = Y_i - \alpha_i(\theta)$, é a função quasi-escore dada por $U_Q(\theta) = \sum_{i=1}^n [Y_i - \alpha_i(\theta)] \frac{\partial \alpha_i(\theta)/\partial \theta}{\sigma_i^2(\theta)}$ [Desmond and Godambe, 1998, 2005].

Assim, temos $E[Y_i] = m_i \mu_i$ e $V(Y_i) = \phi m_i \mu_i (1 - \mu_i)$. A função é:

$$\begin{aligned}
 U_Q(\theta) &= \sum_{i=1}^n \left(\frac{\partial \alpha_i(\theta)}{\partial \theta} \right)^\top \frac{Y_i - m_i \mu_i}{V(Y_i)} \\
 U_{Q,j}(\theta) &= \sum_{i=1}^n \left(\frac{\partial m_i \mu_i}{\partial \theta_j} \right) \frac{Y_i - m_i \mu_i}{V(Y_i)} = \sum_{i=1}^n m_i \left(\frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \theta_j} \right) \frac{Y_i - m_i \mu_i}{\phi m_i \mu_i (1 - \mu_i)} = \sum_{i=1}^n [m_i \mu_i (1 - \mu_i) \cdot x_{ij}] \frac{Y_i - m_i \mu_i}{\phi m_i \mu_i (1 - \mu_i)} \\
 U_{Q,j}(\theta) &= \frac{1}{\phi} \sum_{i=1}^n (Y_i - m_i \mu_i) x_{ij} \\
 U_Q(\theta) &= \frac{1}{\phi} \sum_{i=1}^n (Y_i - m_i \mu_i) x_i, \quad \text{que também é não viesado porque } \mathbb{E}[U_Q(\theta)] = \frac{1}{\phi} \mathbb{E}[U(\theta)] = 0
 \end{aligned}$$

Pela teoria de MLG, sabemos que $\hat{\phi} = \frac{1}{n-p} \sum_{i=1}^n \frac{(Y_i - m_i \hat{\mu}_i)^2}{m_i \hat{\mu}_i (1 - \hat{\mu}_i)}$, onde $p = \dim(\theta)$ é o número de parâmetros estimados. Além da função de estimação U_Q apresentada acima, apresento duas outras, uma (U_W) parecida com U_Q mas dividindo pela variância. Esta é baseada nas ideias apresentadas em Wedderburn [1974], Liang and Zeger [1986]

$$U_W(\theta) = \sum_{i=1}^n \frac{(Y_i - m_i \mu_i) x_i}{V(Y_i)}, \quad V(Y_i) = \hat{\phi} \cdot m_i \mu_i (1 - \mu_i),$$

e outra U_H , função de estimação de Huber, baseada nas ideias apresentadas por Huber [1992] e mais precisamente para o caso de MLG em Cantoni and Ronchetti [2001]

$$U_H(\theta) = \sum_{i=1}^n x_i \cdot \max \left[-c\sqrt{\hat{\phi}m_i\mu_i(\theta)(1-\mu_i(\theta))}, \min \left(c\sqrt{\hat{\phi}m_i\mu_i(\theta)(1-\mu_i(\theta))}, Y_i - m_i\mu_i(\theta) \right) \right],$$

onde c , a constante de ajuste, é tipicamente escolhida de modo a garantir um determinado nível de eficiência assintótica [Cantoni and Ronchetti, 2001].

Nota: Em todas funções que definí não dividi por n , como aparece nas notas de aula, mas poderia e pelas propriedades das funções de estimação se tornam equivalentes pois ao igualar a zero, a equação de estimação resultante resultaria em mesmas estimativas com ou sem n no denominador.

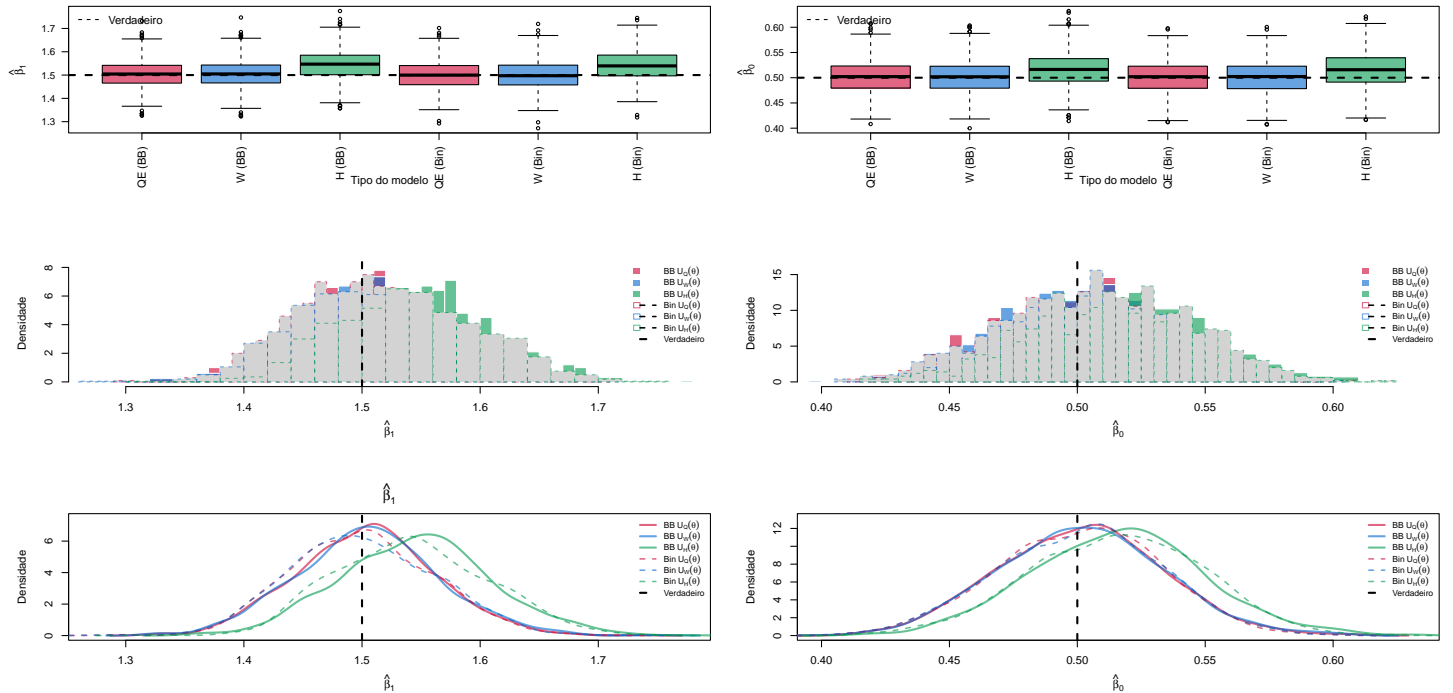
⇒ Parte numérica

Table 2: Resumo dos estimadores para diferentes funções de estimação. Os valores considerados verdadeiros durante a estimação foram $\beta_0 = 0.5$ e $\beta_1 = 1.5$.

Modelo	Função de estimação	N	$\hat{\beta}_0$	SD $\hat{\beta}_0$	$\hat{\beta}_1$	SD $\hat{\beta}_1$
Beta-Binomial (BB)	$U_Q(\theta)$	1000	0.501	0.0320	1.50	0.0588
Beta-Binomial (BB)	$U_W(\theta)$	1000	0.502	0.0324	1.51	0.0607
Beta-Binomial (BB)	$U_H(\theta)$	1000	0.516	0.0337	1.55	0.0637
Binomial	$U_Q(\theta)$	1000	0.501	0.0314	1.50	0.0592
Binomial	$U_W(\theta)$	1000	0.501	0.0320	1.50	0.0610
Binomial	$U_H(\theta)$	1000	0.515	0.0336	1.54	0.0637

A Tabela 2 indica que todos os três estimadores são acurados (não viciados), pois as estimadas para $\hat{\beta}_0$ e $\hat{\beta}_1$ permanecem muito próximas dos valores considerados verdadeiros (0.5 e 1.5) em ambos os cenários, Binomial (sem perturbação) e Beta-Binomial (perturbado). No entanto as funções de estimação U_Q e U_W mostraram eficiência.

Figure 3: Avaliação comparativa do desempenho dos estimadores $U_Q(\theta)$, $U_W(\theta)$ e $U_H(\theta)$ sob os modelos Binomial e Beta-Binomial, obtida por simulação de Monte Carlo (1000 repetições). QE é função de estimação $U_Q(\theta)$, W é função de estimação $U_W(\theta)$ e H é função de estimação $U_H(\theta)$. BB é Beta-Binomial e Bin é Binomial.



A Figura 3, por sua vez, revela a diferença crucial na precisão e robustez: enquanto nos dados Binomiais (linhas tracejadas) os três métodos (U_Q , U_W , U_H) têm distribuições quase idênticas, no cenário perturbado Beta-Binomial

(áreas preenchidas), as distribuições dos estimadores U_Q e U_W tornam-se visivelmente mais próximas à realidade, indicando que essas funções de estimação são robustas mas U_Q foi melhor. Em contraste, a distribuição do estimador robusto de Huber U_H , foi visivelmente mais larga e dispersa, indicando uma grande perda de precisão.

Nota: Era esperado que U_H fosse o melhor, mas como ele depende do valor c escolhido pode-se ter resultados diferentes para cada c .

⇒ Código exercício 1

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(ggplot2, dplyr, tidyr, patchwork)

set.seed(42)
# Parâmetros
n <- 100
beta_0_true <- 0.5
beta_1_true <- 0.4 # H1 é verdadeira
# Gerando dados
x <- rpois(n, 5) # por ser dados de contagem
log_mu_true <- beta_0_true + beta_1_true * x
mu_true <- exp(log_mu_true)
y <- rpois(n, lambda = mu_true)

df <- data.frame(x, y, mu_true)
# Ajuste dos modelos
m_completo <- glm(y ~ x, data = df, family = poisson)
m_nulo <- glm(y ~ 1, data = df, family = poisson)
# Extrair parâmetros
beta0_hat <- coef(m_completo)[1]
beta1_hat <- coef(m_completo)[2]
beta00_hat <- coef(m_nulo)[1]
# log verossimilhanças
logLik_completo <- logLik(m_completo)
logLik_nulo <- logLik(m_nulo)
t_obs = 2 * (1(theta_hat) - 1(theta_hat_0))
t_obs <- 2 * (as.numeric(logLik_completo) - as.numeric(logLik_nulo))
# Valor-p Aproximado (p_a)
p_valor_aprox <- pchisq(t_obs, df = 1, lower.tail = FALSE)
# Gráficos
p1 <- ggplot(df, aes(x = y)) +
  geom_histogram(aes(y = ..density..), bins = 15, fill = "steelblue", alpha = 0.7) +
  geom_density(color = "darkred", linewidth = 1) +
  labs(title = "", x = "Número de nascidos vivos (Y)", y = "Densidade") +
  theme_minimal()

df$y_predito <- predict(m_completo, type = "response")
df$y_predito2 <- predict(m_nulo, type = "response")

p2 <- ggplot(df, aes(x = x, y = y)) +
  geom_point(aes(color = "Dados Observados (y)"), alpha = 0.7, size = 2) +
  geom_line(aes(y = y_predito, color = "Ajuste do Modelo Completo"), linewidth = .5) +
  geom_line(aes(y = y_predito2, color = "Ajuste do Modelo sob H0"), linewidth = .5) +
  geom_line(aes(y = mu_true, color = "Linha Verdadeira"), linetype = "dashed", linewidth = .5) +
  labs(title = "", x = "Experiência do Médico (x)", y = "Número de Nascidos Vivos (y)",
    color = "Legenda:")
  ) +
  scale_color_manual(values = c("Dados Observados (y)" = "grey50",
    "Ajuste do Modelo Completo" = "blue", "Linha Verdadeira" = "red",
    "Ajuste do Modelo sob H0" = "black"
  )) +
  theme_bw() +
  theme(
    legend.position = "bottom", plot.title = element_text(size = 16, hjust = 0.5),
    axis.title = element_text(size = 12)
  )

(p1 | p2) +
  plot_annotation(tag_levels = 'A')

#===SIMULAÇÕES DE MONTE CARLO
set.seed(123)
R <- 5000 # Número de réplicas
```

```

n_sim <- 100      Tamanho da amostra por réplica
alpha <- 0.05    Nível de significância
Parâmetros
beta_0_sim <- 1.0
beta_1_h0 <- 0.0
beta_1_h1 <- 0.2
Vetores para resultados
p_valores_h0 <- numeric(R)
beta1_estimates_h0 <- numeric(R)
p_valores_h1 <- numeric(R)
beta1_estimates_h1 <- numeric(R)
pb <- txtProgressBar(min = 0, max = R, style = 3)

for (i in 1:R) {
  x_sim <- rpois(n_sim, 5)
  Sob H0 (beta_1 = 0)
  lambda_h0 <- exp(beta_0_sim + beta_1_h0 * x_sim)
  y_h0 <- rpois(n_sim, lambda = lambda_h0)

  modelo_h0 <- glm(y_h0 ~ x_sim, family = poisson)
  modelo_null_h0 <- glm(y_h0 ~ 1, family = poisson)
  Teste de Razão de Verossimilhança
  t_obs_h0 <- 2 * (logLik(modelo_h0) - logLik(modelo_null_h0))
  p_valores_h0[i] <- pchisq(t_obs_h0, df = 1, lower.tail = FALSE)
  beta1_estimates_h0[i] <- coef(modelo_h0)[2]
  Sob H1 (beta_1 = 0.2)
  lambda_h1 <- exp(beta_0_sim + beta_1_h1 * x_sim)
  y_h1 <- rpois(n_sim, lambda = lambda_h1)

  modelo_h1 <- glm(y_h1 ~ x_sim, family = poisson)
  modelo_null_h1 <- glm(y_h1 ~ 1, family = poisson)
  Teste de Razão de Verossimilhança (LRT)
  t_obs_h1 <- 2 * (logLik(modelo_h1) - logLik(modelo_null_h1))
  p_valores_h1[i] <- pchisq(t_obs_h1, df = 1, lower.tail = FALSE)
  beta1_estimates_h1[i] <- coef(modelo_h1)[2]

  setTxtProgressBar(pb, i)
}
close(pb)

erro_tipo_I <- mean(p_valores_h0 < alpha)
poder_teste <- mean(p_valores_h1 < alpha)

df_resultados <- data.frame(
  p_valor = c(p_valores_h0, p_valores_h1),
  beta1_est = c(beta1_estimates_h0, beta1_estimates_h1),
  Hipotese = rep(c("H0 (Nula)", "H1 (Alternativa)"), each = R)
)

Gráficos
Ho
plot_pvals_h0 <- ggplot(filter(df_resultados, Hipotese == "H0 (Nula)"), aes(x = p_valor)) +
  geom_histogram(aes(y = ..density..), binwidth = 0.05, color = "white",
    fill = "darkblue", boundary = 0) +
  geom_hline(yintercept = 1, color = "red", linetype = "dashed", linewidth = 1) +
  labs(title = "",
    subtitle = "",
    x = "Valor-p",
    y = "Densidade") +
  scale_x_continuous(breaks = seq(0, 1, 0.2)) +
  coord_cartesian(xlim = c(0, 1)) +
  theme_minimal()

Gráfico para H1
plot_pvals_h1 <- ggplot(filter(df_resultados, Hipotese == "H1 (Alternativa)"),

```

```

aes(x = p_valor)) +
  geom_histogram(binwidth = 0.05, color = "white", fill = "darkred", boundary = 0) +
  labs(title = "",
        subtitle = "",
        x = "Valor-p",
        y = "Contagem") +
  scale_x_continuous(breaks = seq(0, 1, 0.2)) +
  coord_cartesian(xlim = c(0, 1)) +
  theme_minimal()
plot_p_valores_combinado <- (plot_pvals_h0 | plot_pvals_h1)

plot_beta_estes <- ggplot(df_resultados, aes(x = beta1_est, fill = Hipotese)) +
  geom_histogram(binwidth = 0.02, color = "white", alpha = 0.7, position = "identity") +
  labs(
    title = "",
    x = expression("Valor Estimado de"~hat(beta)[1]),
    y = "Contagem"
  ) +
  geom_vline(xintercept = beta_1_h0, color = "darkblue", linetype = "dashed", linewidth = 1) +
  geom_vline(xintercept = beta_1_h1, color = "darkred", linetype = "dashed", linewidth = 1) +
  scale_fill_manual(values = c("H0 (Nula)" = "darkblue", "H1 (Alternativa)" = "darkred")) +
  theme_minimal() +
  guides(fill = guide_legend(title = "")) +
  annotate("text", x = beta_1_h0 - 0.05, y = 400, label = "", color = "darkblue", hjust = 1) +
  annotate("text", x = beta_1_h1 + 0.05, y = 400, label = "", color = "darkred", hjust = 0)

((plot_pvals_h0 + plot_pvals_h1) / plot_beta_estes) +
  plot_annotation(tag_levels = 'A')

```

⇒ Código R Exercício 3 (Numérico e Simulação)

```

library(tidyverse)
#  $\mu_i(\theta) = \exp(x_i' \cdot \theta) / (1 + \exp(x_i' \cdot \theta))$ 
mu_stable <- function(params, Xmat) {
  eta <- Xmat %*% params
  # Prender eta em limites razoáveis para evitar Inf/NaN em exp(eta)
  eta <- pmax(-700, pmin(700, eta))
  mu <- exp(eta) / (1 + exp(eta))
  return(as.vector(mu))
}
#  $V_i(\theta) = m_i \cdot \mu_i \cdot (1 - \mu_i)$ 
V_fun_stable <- function(params, Xmat, m) {
  mu_i <- mu_stable(params, Xmat)
  V_i <- m * mu_i * (1 - mu_i)
  V_i <- V_i + 1e-10
  return(as.vector(V_i))
}
#
rbetabinom <- function(n, size, mu, phi) {
  if (phi <= 1) {
    return(rbinom(n, size = size, prob = mu))
  }
  nu <- ((size - 1) / (phi - 1)) - 1
  if (nu <= 0) {
    return(rbinom(n, size = size, prob = mu))
  }
  alpha_i <- mu * nu
  beta_i <- (1 - mu) * nu
  p_from_beta <- rbeta(n, shape1 = alpha_i, shape2 = beta_i)
  y <- rbinom(n, size = size, prob = p_from_beta)
  return(y)
}
#
calculate_phi_chi2 <- function(Y, m_val, params, Xmat, p) {
  n <- length(Y)

```

```

E_Y_i <- m_val * mu_stable(params, Xmat)
V_i_no_phi <- V_fun_stable(params, Xmat, m_val)

pearson_resids_sq <- ((Y - E_Y_i)^2) / V_i_no_phi
phi_hat <- sum(pearson_resids_sq) / (n - p)
return(phi_hat)
}

# Huber
psi_c <- function(x, c_val) {
  pmax(-c_val, pmin(c_val, x))
}

#Função de Estimação: Quasi-Escore (U_QE)
estimating_function_QE <- function(params, Y, Xmat, m) {
  mu_i <- mu_stable(params, Xmat)
  residual <- Y - (m * mu_i)
  g_vector <- t(Xmat) %*% residual
  return(as.vector(g_vector))
}

f_Q_QE <- function(params, Y, Xmat, m) {
  g_vec <- estimating_function_QE(params, Y, Xmat, m)
  Q <- sum(g_vec^2)
  return(Q)
}

# Função de Estimação: Ponderada (U_W)
estimating_function_W <- function(params, Y, Xmat, m, phi_hat) {
  mu_i <- mu_stable(params, Xmat)
  V_i <- V_fun_stable(params, Xmat, m)
  residual <- Y - (m * mu_i)
  weighted_resid <- residual / (phi_hat * V_i)
  g_vector <- t(Xmat) %*% weighted_resid
  return(as.vector(g_vector))
}

f_Q_W <- function(params, Y, Xmat, m, phi_hat) {
  g_vec <- estimating_function_W(params, Y, Xmat, m, phi_hat)
  Q <- sum(g_vec^2)
  return(Q)
}

#Função de Estimação: Huber (U_H)
estimating_function_H <- function(params, Y, Xmat, m, phi_hat, c_huber) {
  mu_i <- mu_stable(params, Xmat)
  E_Y_i <- m * mu_i

  V_i_star <- phi_hat * V_fun_stable(params, Xmat, m)
  sd_i_star <- sqrt(V_i_star)

  r_std <- (Y - E_Y_i) / sd_i_star
  psi_r <- psi_c(r_std, c_huber)
  huber_term <- psi_r * sd_i_star

  g_vector <- t(Xmat) %*% huber_term
  return(as.vector(g_vector))
}

f_Q_H <- function(params, Y, Xmat, m, phi_hat, c_huber) {
  g_vec <- estimating_function_H(params, Y, Xmat, m, phi_hat, c_huber)
  Q <- sum(g_vec^2)
  return(Q)
}

#
set.seed(42)
R <- 1000
n <- 500
m <- 10

```

```

theta_true <- c(beta_0 = 0.5, beta_1 = 1.5)
phi_true <- 15.0 #PERTURBAÇÃO
c_huber_val <- 1.345

results_list <- list()
initial_params <- c(0, 0)
p_model <- length(initial_params)

for (i in 1:R) {
  x <- runif(n, -1, 1)
  Xmat <- model.matrix(~x)
  mu_i_true <- mu_stable(theta_true, Xmat)
  #Cenário 1: Dados Binomiais (Modelo Correto)
  y_bin <- rbinom(n, size = m, prob = mu_i_true)

  fit_qe_bin <- optim(
    par = initial_params, fn = f_Q_QE,
    Y = y_bin, Xmat = Xmat, m = m, method = "BFGS"
  )
  phi_hat_bin <- calculate_phi_chi2(y_bin, m, fit_qe_bin$par, Xmat, p_model)
  #
  fit_w_bin <- optim(
    par = initial_params, fn = f_Q_W,
    Y = y_bin, Xmat = Xmat, m = m, method = "BFGS",
    phi_hat = phi_hat_bin
  )
  #
  fit_h_bin <- optim(
    par = initial_params, fn = f_Q_H,
    Y = y_bin, Xmat = Xmat, m = m, method = "Nelder-Mead",
    phi_hat = phi_hat_bin, c_huber = c_huber_val
  )
  results_list[[paste0(i, "_qe_bin")]] <- c(fit_qe_bin$par,
    Estimator = "QE", Data = "Binomial")
  results_list[[paste0(i, "_w_bin")]] <- c(fit_w_bin$par,
    Estimator = "Ponderada (W)", Data = "Binomial")
  results_list[[paste0(i, "_h_bin")]] <- c(fit_h_bin$par,
    Estimator = "Huber (H)", Data = "Binomial")

  # Dados Beta-Binomiais (Perturbado)
  y_perturbed <- rbetabinom(n, size = m, mu = mu_i_true, phi = phi_true)
  #
  fit_qe_pert <- optim(
    par = initial_params, fn = f_Q_QE,
    Y = y_perturbed, Xmat = Xmat, m = m, method = "BFGS"
  )
  phi_hat_pert <- calculate_phi_chi2(y_perturbed, m, fit_qe_pert$par, Xmat, p_model)

  fit_w_pert <- optim(
    par = initial_params, fn = f_Q_W,
    Y = y_perturbed, Xmat = Xmat, m = m, method = "BFGS",
    phi_hat = phi_hat_pert
  )
  fit_h_pert <- optim(
    par = initial_params, fn = f_Q_H,
    Y = y_perturbed, Xmat = Xmat, m = m, method = "Nelder-Mead",
    phi_hat = phi_hat_pert, c_huber = c_huber_val
  )
  results_list[[paste0(i, "_qe_pert")]] <- c(fit_qe_pert$par,
    Estimator = "QE", Data = "Beta-Binomial (Pert.)")
  results_list[[paste0(i, "_w_pert")]] <- c(fit_w_pert$par,
    Estimator = "Ponderada (W)", Data = "Beta-Binomial (Pert.)")
  results_list[[paste0(i, "_h_pert")]] <- c(fit_h_pert$par,
    Estimator = "Huber (H)", Data = "Beta-Binomial (Pert.)")
}

```

```

  if (i %% 100 == 0) cat("Replicação", i, "/", R, "concluída...\n")
}
#
all_results_df <- do.call(rbind, results_list)
all_results <- data.frame(
  beta_0 = as.numeric(all_results_df[, 1]),
  beta_1 = as.numeric(all_results_df[, 2]),
  Estimator = all_results_df[, 3],
  Data = all_results_df[, 4]
)
#
all_results$Estimator_Short <- all_results$Estimator
all_results$Estimator_Short <- gsub("QE", "QE", all_results$Estimator_Short)
all_results$Estimator_Short <- gsub("Ponderada \\(W\\)", "W", all_results$Estimator_Short)
all_results$Estimator_Short <- gsub("Huber \\(H\\)", "H", all_results$Estimator_Short)
all_results$Data_Short <- all_results$Data
all_results$Data_Short <- gsub("Beta-Binomial \\(Pert.\\)", "(BB)", all_results$Data_Short)
all_results$Data_Short <- gsub("Binomial", "(Bin)", all_results$Data_Short)
#
all_results$Model <- paste(all_results$Estimator_Short, all_results$Data_Short)
#
all_results$Estimator <- factor(all_results$Estimator,
levels = c("QE", "Ponderada (W)", "Huber (H)"))
all_results <- all_results[order(all_results$Data,
all_results$Estimator), ]
all_results$Model <- factor(all_results$Model,
levels = unique(all_results$Model))
#
summary_stats <- all_results %>%
  group_by(Data, Estimator) %>%
  summarise(
    N = n(),
    Mean_b0 = mean(beta_0),
    Median_b0 = median(beta_0),
    SD_b0 = sd(beta_0),
    Mean_b1 = mean(beta_1),
    Median_b1 = median(beta_1),
    SD_b1 = sd(beta_1),
    .groups = 'drop'
  )
print(summary_stats)
#
col_qe <- rgb(0.8, 0, 0.2, alpha = 0.6)
col_w <- rgb(0, 0.4, 0.8, alpha = 0.6)
col_h <- rgb(0, 0.6, 0.3, alpha = 0.6)
#
par(mfrow = c(3, 2), mar = c(6.5, 4.5, 3, 2))
# Boxplot para beta_1
boxplot(beta_1 ~ Model, data = all_results,
  las = 2,
  col = rep(c(col_qe, col_w, col_h), 2),
  main = "",
  ylab = expression(hat(beta)[1]),
  xlab = "Tipo do modelo",
  xaxt = "n",
  cex.axis = 0.9)
#
axis(1, at = 1:6, labels = levels(all_results$Model), las = 2)
abline(h = theta_true[2], col = "black", lwd = 2, lty = 2)
legend("topleft", legend = "Verdadeiro", lty = 2, bty = "n")
# Boxplot para beta_0
boxplot(beta_0 ~ Model, data = all_results,
  las = 2,

```

```

col = rep(c(col_qe, col_w, col_h), 2),
main = "",
ylab = expression(hat(beta)[0]),
xlab = "Tipo do modelo",
xast = "n",
cex.axis = 0.9)
axis(1, at = 1:6, labels = levels(all_results$Model), las = 2)
abline(h = theta_true[1], col = "black", lwd = 2, lty = 2)
legend("topleft", legend = "Verdadeiro", lty = 2, bty = "n")
# Histograma para beta_1
h_qe_p <- hist(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "QE")$beta_1, plot = FALSE, breaks = 30)
h_w_p <- hist(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "W")$beta_1, plot = FALSE, breaks = 30)
h_h_p <- hist(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "H")$beta_1, plot = FALSE, breaks = 30)
h_qe_b <- hist(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "QE")$beta_1, plot = FALSE, breaks = 30)
h_w_b <- hist(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "W")$beta_1, plot = FALSE, breaks = 30)
h_h_b <- hist(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "H")$beta_1,
plot = FALSE, breaks = 30)
xlims <- range(all_results$beta_1)
ylims <- range(0, h_qe_p$density, h_w_p$density, h_h_p$density,
h_qe_b$density, h_w_b$density, h_h_b$density) * 1.05
plot(h_qe_p, freq = FALSE, col = col_qe, xlim = xlims, ylim = ylims, border = "white",
main = "",
xlab = expression(hat(beta)[1]), ylab = "Densidade")
plot(h_w_p, freq = FALSE, col = col_w, add = TRUE, border = "white")
plot(h_h_p, freq = FALSE, col = col_h, add = TRUE, border = "white")
plot(h_qe_b, freq = FALSE, border = col_qe, lty = 2, lwd = 1.5, add = TRUE)
plot(h_w_b, freq = FALSE, border = col_w, lty = 2, lwd = 1.5, add = TRUE)
plot(h_h_b, freq = FALSE, border = col_h, lty = 2, lwd = 1.5, add = TRUE)
abline(v = theta_true[2], col = "black", lwd = 2, lty = 2)
legend("topright",
legend = c(expression("BB"~U[Q] (theta)), expression("BB"~U[W] (theta)),
expression("BB"~U[H] (theta)), expression("Bin"~U[Q] (theta)),
expression("Bin"~U[W] (theta)), expression("Bin"~U[H] (theta)), "Verdadeiro"),
fill = c(col_qe, col_w, col_h, NA, NA, NA, NA),
border = c(NA, NA, NA, col_qe, col_w, col_h, NA),
lty = c(NA, NA, NA, 2, 2, 2, 2),
lwd = c(NA, NA, NA, 1.5, 1.5, 1.5, 2),
bty = "n", cex = 0.8)
# Histograma para beta_0
h_qe_p <- hist(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "QE")$beta_0, plot = FALSE, breaks = 30)
h_w_p <- hist(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "W")$beta_0, plot = FALSE, breaks = 30)
h_h_p <- hist(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "H")$beta_0, plot = FALSE, breaks = 30)
h_qe_b <- hist(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "QE")$beta_0, plot = FALSE, breaks = 30)
h_w_b <- hist(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "W")$beta_0, plot = FALSE, breaks = 30)
h_h_b <- hist(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "H")$beta_0, plot = FALSE, breaks = 30)
xlims <- range(all_results$beta_0)
ylims <- range(0, h_qe_p$density, h_w_p$density,
h_h_p$density, h_qe_b$density, h_w_b$density, h_h_b$density) * 1.05
plot(h_qe_p, freq = FALSE, col = col_qe, xlim = xlims, ylim = ylims, border = "white",
main = "",
xlab = expression(hat(beta)[0]), ylab = "Densidade")
plot(h_w_p, freq = FALSE, col = col_w, add = TRUE, border = "white")

```



```

plot(h_h_p, freq = FALSE, col = col_h, add = TRUE, border = "white")
plot(h_qe_b, freq = FALSE, border = col_qe, lty = 2, lwd = 1.5, add = TRUE)
plot(h_w_b, freq = FALSE, border = col_w, lty = 2, lwd = 1.5, add = TRUE)
plot(h_h_b, freq = FALSE, border = col_h, lty = 2, lwd = 1.5, add = TRUE)
abline(v = theta_true[1], col = "black", lwd = 2, lty = 2)
legend("topright",
      legend = c(expression("BB"~U[Q] (theta)), expression("BB"~U[W] (theta)),
        expression("BB"~U[H] (theta)), expression("Bin"~U[Q] (theta)), expression("Bin"~U[W] (theta)),
        expression("Bin"~U[H] (theta)), "Verdadeiro"),
      fill = c(col_qe, col_w, col_h, NA, NA, NA, NA),
      border = c(NA, NA, NA, col_qe, col_w, col_h, NA),
      lty = c(NA, NA, NA, 2, 2, 2, 2),
      lwd = c(NA, NA, NA, 1.5, 1.5, 1.5, 2),
      bty = "n", cex = 0.8)

# Densidades para beta_1
d1_qe_p <- density(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "QE")$beta_1)
d1_w_p <- density(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "W")$beta_1)
d1_h_p <- density(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "H")$beta_1)
d1_qe_b <- density(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "QE")$beta_1)
d1_w_b <- density(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "W")$beta_1)
d1_h_b <- density(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "H")$beta_1)
xlims <- range(all_results$beta_1)
ylims <- range(0, d1_qe_p$y, d1_w_p$y, d1_h_p$y, d1_qe_b$y, d1_w_b$y, d1_h_b$y) * 1.05
plot(NA, xlim = xlims, ylim = ylims,
      main = expression(hat(beta)[1]),
      xlab = expression(hat(beta)[1]), ylab = "Densidade")
lines(d1_qe_p, col = col_qe, lwd = 2, lty = 1)
lines(d1_w_p, col = col_w, lwd = 2, lty = 1)
lines(d1_h_p, col = col_h, lwd = 2, lty = 1)
lines(d1_qe_b, col = col_qe, lwd = 1.5, lty = 2)
lines(d1_w_b, col = col_w, lwd = 1.5, lty = 2)
lines(d1_h_b, col = col_h, lwd = 1.5, lty = 2)
abline(v = theta_true[2], col = "black", lwd = 2, lty = 2)
legend("topright",
      legend = c(expression("BB"~U[Q] (theta)), expression("BB"~U[W] (theta)),
        expression("BB"~U[H] (theta)), expression("Bin"~U[Q] (theta)), expression("Bin"~U[W] (theta)),
        expression("Bin"~U[H] (theta)), "Verdadeiro"),
      col = c(col_qe, col_w, col_h, col_qe, col_w, col_h, "black"),
      lwd = c(2, 2, 2, 1.5, 1.5, 1.5, 2),
      lty = c(1, 1, 1, 2, 2, 2, 2),
      bty = "n", cex = 0.8)

# Densidades para beta_0
d0_qe_p <- density(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "QE")$beta_0)
d0_w_p <- density(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "W")$beta_0)
d0_h_p <- density(subset(all_results,
Data_Short == "(BB)" & Estimator_Short == "H")$beta_0)
d0_qe_b <- density(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "QE")$beta_0)
d0_w_b <- density(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "W")$beta_0)
d0_h_b <- density(subset(all_results,
Data_Short == "(Bin)" & Estimator_Short == "H")$beta_0)
xlims <- range(all_results$beta_0)
ylims <- range(0, d0_qe_p$y, d0_w_p$y, d0_h_p$y, d0_qe_b$y, d0_w_b$y, d0_h_b$y) * 1.05
plot(NA, xlim = xlims, ylim = ylims,
      main = "",

```



```

xlab = expression(hat(beta)[0]), ylab = "Densidade")
lines(d0_qe_p, col = col_qe, lwd = 2, lty = 1)
lines(d0_w_p, col = col_w, lwd = 2, lty = 1)
lines(d0_h_p, col = col_h, lwd = 2, lty = 1)
lines(d0_qe_b, col = col_qe, lwd = 1.5, lty = 2)
lines(d0_w_b, col = col_w, lwd = 1.5, lty = 2)
lines(d0_h_b, col = col_h, lwd = 1.5, lty = 2)
abline(v = theta_true[1], col = "black", lwd = 2, lty = 2)
legend("topright",
      legend = c(expression("BB"~U[Q] (theta)), expression("BB"~U[W] (theta)),
        expression("BB"~U[H] (theta)), expression("Bin"~U[Q] (theta)), expression("Bin"~U[W] (theta)),
        expression("Bin"~U[H] (theta)), "Verdadeiro"),
      col = c(col_qe, col_w, col_h, col_qe, col_w, col_h, "black"),
      lwd = c(2, 2, 2, 1.5, 1.5, 1.5, 2),
      lty = c(1, 1, 1, 2, 2, 2, 2),
      bty = "n", cex = 0.8)
par(mfrow = c(1, 1))

```

References

- Alexandre G Patriota. A classical measure of evidence for general null hypotheses. *Fuzzy Sets and Systems*, 233:74–88, 2013.
- Alexandre G Patriota. Uma medida de evidência alternativa para testar hipóteses gerais. *Ciência e Natura*, 36:14–22, 2014.
- Alexandre Galvao Patriota. On some assumptions of the null hypothesis statistical testing. *Educational and Psychological Measurement*, 77(3):507–528, 2017.
- Mark J Schervish. P values: what they are and what they are not. *The American Statistician*, 50(3):203–206, 1996.
- Ronald L Wasserstein and Nicole A Lazar. The asa statement on p-values: context, process, and purpose, 2016.
- Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.
- Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a non-polynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.
- A. F. Desmond and V. P. Godambe. The theory of estimating functions. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd, Chichester, 2005. doi: <https://2024.sci-hub.se/6550/f7ff339e5be54a0280c89017470376bd/10.1002@9781118445112.stat05847.pdf>.
- Robert WM Wedderburn. Quasi-likelihood functions, generalized linear models, and the gauss—newton method. *Biometrika*, 61(3):439–447, 1974.
- Anthony F Desmond and VP Godambe. Estimating functions. *Encyclopedia of Biostatistics* (eds. P. Armitage and T. Colton), 2:1375–1386, 1998.
- Kung-Yee Liang and Scott L Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1): 13–22, 1986.
- Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.
- Eva Cantoni and Elvezio Ronchetti. Robust inference for generalized linear models. *Journal of the American Statistical Association*, 96(455):1022–1030, 2001.