

```

set.seed(2025)

# a) Poisson: Teste de Lambda = lambda0
n <- 100
lambda_0 <- 5
# Simulação (H0 verdadeira)
dados_pois <- rpois(n, lambda = lambda_0)
lambda_hat <- mean(dados_pois)
# Estatística Z
z_stat <- (lambda_hat - lambda_0) / sqrt(lambda_0/n)
p_val <- 2 * (1 - pnorm(abs(z_stat)))
cat("a) Poisson\nEMV:", lambda_hat, "| Z-stat:", z_stat, "| p-valor:", p_val)

## a) Poisson
## EMV: 5.17 | Z-stat: 0.7602631 | p-valor: 0.4470973

# b) Binomial: Teste p = 0.5 e Poder
n_bin <- 50 # tamanho da amostra (ensaios)
p_0 <- 0.5
# Simulação (H0 verdadeira)
x_bin <- rbinom(1, n_bin, p_0) # número de sucessos
p_hat <- x_bin / n_bin
# Estatística Z
z_bin <- (p_hat - p_0) / sqrt((p_0 * (1-p_0)) / n_bin)
p_val_bin <- 2 * (1 - pnorm(abs(z_bin)))
# Cálculo do Poder para p = 0.8 (Simulação)
n_sim <- 1000
p_alt <- 0.8
rejeicoes <- 0
for(i in 1:n_sim){
  x_sim <- rbinom(1, n_bin, p_alt)
  p_sim <- x_sim / n_bin
  z_sim <- (p_sim - p_0) / sqrt((p_0 * (1-p_0)) / n_bin)
  if(abs(z_sim) > 1.96) rejeicoes <- rejeicoes + 1}
poder <- rejeicoes / n_sim
cat("b) Binomial\np-chapéu:", p_hat, "| Z-stat:", z_bin, "| p-valor:", p_val_bin,
    "\nPoder do teste para p=0.8 (via simulação):", poder)

## b) Binomial
## p-chapéu: 0.52 | Z-stat: 0.2828427 | p-valor: 0.7772974
## Poder do teste para p=0.8 (via simulação): 0.999

# c) Exponencial: Teste da Razão de Verossimilhança (TRV)
lambda_0_exp <- 2
dados_exp <- rexp(n, rate = lambda_0_exp) # H0 verdadeira
lambda_hat_exp <- 1/mean(dados_exp)
# Estatística TRV (Deviance)
# TRV = 2n [ (lambda_0/lambda_hat) - 1 - log(lambda_0/lambda_hat) ]
ratio <- lambda_0_exp / lambda_hat_exp
trv_stat <- 2 * n * (ratio - 1 - log(ratio))
p_val_trv <- 1 - pchisq(trv_stat, df = 1)
cat("c) Exponencial (TRV)\nEMV:", lambda_hat_exp, "| TRV-stat:", trv_stat,
    "| p-valor:", p_val_trv)

```

```
## c) Exponencial (TRV)
## EMV: 1.937436 | TRV-stat: 0.1020855 | p-valor: 0.7493412
```

```
# d) Normal: Teste Escore para Sigma = 1
# Função para realizar o teste escore
teste_escore_sigma <- function(dados, sigma0 = 1){
  n <- length(dados)
  mu_hat <- mean(dados)
  S2_biased <- sum((dados - mu_hat)^2) # Numerador da variância
  # Estatística derivada
  score_stat <- (S2_biased - n*sigma0^2)^2 / (2 * n * sigma0^4)
  p_val <- 1 - pchisq(score_stat, df = 1)
  return(p_val)}
# Estudo de simulação (Verificar Taxa de Erro Tipo I)
rejeicoes_norm <- 0 # Erro Tipo I
nao_rejeicoes_alt <- 0 # Erro Tipo II
for(i in 1:n_sim){
  # --- Erro Tipo I ---
  dados_norm <- rnorm(n, mean = 10, sd = 1) # H0 verdadeira
  if(teste_escore_sigma(dados_norm) < 0.05){
    rejeicoes_norm <- rejeicoes_norm + 1}
  # --- Erro Tipo II ---
  dados_alt <- rnorm(n, mean = 10, sd = 1.2) # H1 verdadeira (sd != 1)
  if(teste_escore_sigma(dados_alt) >= 0.05){
    nao_rejeicoes_alt <- nao_rejeicoes_alt + 1}}
erro_tipo1 <- rejeicoes_norm / n_sim
erro_tipo2 <- nao_rejeicoes_alt / n_sim
cat("d) Normal (Teste Escore)\nTaxa de Erro Tipo I (alvo 0.05):", erro_tipo1,
    "\nTaxa de Erro Tipo II:", erro_tipo2, "\n")
```

```
## d) Normal (Teste Escore)
## Taxa de Erro Tipo I (alvo 0.05): 0.039
## Taxa de Erro Tipo II: 0.218
```

```
# e) Tweedie: Teste Wald para Phi = 1
# Configuração Tweedie
mu <- 10; phi_0 <- 1; p <- 1.5
# Função para teste Wald
teste_wald_phi <- function(dados, p_known){
  n <- length(dados)
  mu_hat <- mean(dados)
  # Estimativa de phi (baseada no Deviance ou Pearson)
  # Usaremos Pearson para simplificar: Phi = sum((y-mu)^2 / mu^p) / (n-1)
  phi_hat <- sum((dados - mu_hat)^2 / (mu_hat^p)) / (n - 1)
  # Variância assintótica de phi_hat aprox 2*phi^2/n (para phi=1 sob H0)
  var_phi <- 2 * (1^2) / n
  wald_stat <- (phi_hat - 1)^2 / var_phi
  p_val <- 1 - pchisq(wald_stat, df = 1)
  return(c(phi_hat, p_val))}
# Estudo de Simulação
rejeicoes_tw <- 0
phis_estimados <- numeric(n_sim)
for(i in 1:n_sim){
```

```

# Gerar dados Tweedie (requer pacote tweedie)
dados_tw <- rtweedie(n, mu = mu, phi = phi_0, power = p)
res <- teste_wald_phi(dados_tw, p)
phis_estimados[i] <- res[1]
if(res[2] < 0.05) rejeicoes_tw <- rejeicoes_tw + 1}
erro_tipo1_tw <- rejeicoes_tw / n_sim
cat("e) Tweedie (Teste Wald)\nMédia Phi estimado:", mean(phis_estimados),
    "\nTaxa de Erro Tipo I (Wald):", erro_tipo1_tw, "\n")

```

```

## e) Tweedie (Teste Wald)
## Média Phi estimado: 0.9967865
## Taxa de Erro Tipo I (Wald): 0.07

```