```r
# Bibliotecas necessárias
if(!require(MASS)) install.packages("MASS")
if(!require(statmod)) install.packages("statmod")
if(!require(tweedie)) install.packages("tweedie")

set.seed(2025)
alpha_conf <- 0.95
z_score <- qnorm(1 - (1-alpha_conf)/2)

# a) Poisson
lambda_true <- 5; n <- 50
x_pois <- rpois(n, lambda_true)

emv_pois <- mean(x_pois)
# IC 1: Wald
se_pois <- sqrt(emv_pois/n)
ic_wald_pois <- c(emv_pois - z_score*se_pois, emv_pois + z_score*se_pois)
# IC 2: Exato
ic_exact_pois <- poisson.test(sum(x_pois), T=n)$conf.int[1:2]

cat("a) Poisson ( Lambda Real =", lambda_true, ")\nEMV:", emv_pois,
    "\nIC Wald:", ic_wald_pois, "\nIC Exato:", ic_exact_pois)
```

```
## a) Poisson ( Lambda Real = 5 )
## EMV: 5.32
## IC Wald: 4.680679 5.959321
## IC Exato: 4.699828 5.999249
```

```r
# b) Binomial
n_trials <- 100; p_true <- 0.4
k_sucessos <- rbinom(1, n_trials, p_true)

emv_p <- k_sucessos / n_trials
# IC 1: Wald
se_bin <- sqrt(emv_p * (1-emv_p) / n_trials)
ic_wald_bin <- c(emv_p - z_score*se_bin, emv_p + z_score*se_bin)
# IC 2: Wilson
ic_wilson <- prop.test(k_sucessos, n_trials, conf.level=alpha_conf)$conf.int[1:2]

cat("b) Binomial ( p Real =", p_true, ")\nEMV:", emv_p,
    "\nIC Wald:", ic_wald_bin, "\nIC Wilson:", ic_wilson)
```

```
## b) Binomial ( p Real = 0.4 )
## EMV: 0.36
## IC Wald: 0.2659217 0.4540783
## IC Wilson: 0.2681721 0.4627255
```

```r
# c) Exponencial
lambda_exp_true <- 2; n <- 50
x_exp <- rexp(n, lambda_exp_true)

emv_exp <- 1/mean(x_exp)
```

```r
# IC 1: Assintótico
se_exp <- emv_exp / sqrt(n)
ic_asy_exp <- c(emv_exp - z_score*se_exp, emv_exp + z_score*se_exp)
# IC 2: Exato
ic_exact_exp <- c(qchisq(0.025, 2*n)/(2*n*mean(x_exp)),
                  qchisq(0.975, 2*n)/(2*n*mean(x_exp)))

cat("c) Exponencial ( Lambda Real =", lambda_exp_true, ")
    \nEMV:", emv_exp, "\nIC Assintótico:", ic_asy_exp, "\nIC Exato:", ic_exact_exp)
```

```
## c) Exponencial ( Lambda Real = 2 )
##
## EMV: 2.465305
## IC Assintótico: 1.78197 3.14864
## IC Exato: 1.829797 3.194079
```

```r
# d) Normal (para mu)
mu_true <- 10; sigma_true <- 2; n <- 50
x_norm <- rnorm(n, mu_true, sigma_true)

emv_mu <- mean(x_norm)
# IC 1: t-Student (Exato)
ic_t <- t.test(x_norm, conf.level=alpha_conf)$conf.int[1:2]

# IC 2: Z-Assintótico (usando SD do EMV)
sd_emv <- sqrt(sum((x_norm - mean(x_norm))^2)/n) # Desvio padrão do EMV (viesado)
ic_z <- c(emv_mu - z_score * (sd_emv/sqrt(n)),
          emv_mu + z_score * (sd_emv/sqrt(n)))

cat("d) Normal ( Mu Real =", mu_true, ")\nEMV:", emv_mu, "
    \nIC t-Student:", ic_t, "\nIC Z-Assintótico:", ic_z)
```

```
## d) Normal ( Mu Real = 10 )
## EMV: 9.593518
##
## IC t-Student: 8.994055 10.19298
## IC Z-Assintótico: 9.014731 10.1723
```

```r
# e) Gama
shape_true <- 2; rate_true <- 1; n <- 100
x_gamma <- rgamma(n, shape=shape_true, rate=rate_true)

fit_gamma <- fitdistr(x_gamma, "gamma")
emv_gamma <- fit_gamma$estimate
# IC 1: Assintótico (Hessiana/Wald)
se_gamma <- fit_gamma$sd
ic_asy_gamma <- emv_gamma[1] + c(-1,1)*z_score*se_gamma[1]

# IC 2: Razão de Verossimilhança (Perfilada) para o parâmetro 'shape'
# Função de log-verossimilhança perfilada para alpha (shape)
loglik_profile <- function(alpha, data) {
  beta_hat <- alpha / mean(data) # EMV de beta dado alpha
```

```r
  sum(dgamma(data, shape = alpha, rate = beta_hat, log = TRUE))
}
max_loglik <- fit_gamma$loglik
cutoff <- max_loglik - qchisq(0.95, 1)/2
# Encontrar raízes onde logL(theta) = logL(hat) - chi/2
lower_prof <- uniroot(function(a) loglik_profile(a, x_gamma) - cutoff,
                      c(0.1, emv_gamma[1]))$root
upper_prof <- uniroot(function(a) loglik_profile(a, x_gamma) - cutoff,
                      c(emv_gamma[1], 10))$root
ic_profile <- c(lower_prof, upper_prof)

cat("e) Gama ( Shape Real =", shape_true, ")
    \nEMV Shape:", emv_gamma[1], "\nIC Assintótico (Wald):", ic_asy_gamma,
    "\nIC Razão de Verossimilhança:", ic_profile)
```

```
## e) Gama ( Shape Real = 2 )
##
## EMV Shape: 2.234418
## IC Assintótico (Wald): 1.655343 2.813492
## IC Razão de Verossimilhança: 1.706975 2.867686
```

```r
# f) Log-Normal (para mu log)
meanlog_true <- 1; n <- 100
x_lnorm <- rlnorm(n, meanlog=meanlog_true, sdlog=0.5)
log_x <- log(x_lnorm)

emv_mulog <- mean(log_x)
# IC 1: t-Student nos logs
ic_t_log <- t.test(log_x)$conf.int[1:2]

# IC 2: Z-Assintótico nos logs (para n grande)
se_log <- sd(log_x) / sqrt(n)
ic_z_log <- c(emv_mulog - z_score*se_log, emv_mulog + z_score*se_log)

cat("f) Log-Normal ( Mu Log Real =", meanlog_true, ")
    \nEMV:", emv_mulog, "\nIC t (logs):", ic_t_log, "\nIC Z-Assintótico:", ic_z_log)
```

```
## f) Log-Normal ( Mu Log Real = 1 )
##
## EMV: 1.064833
## IC t (logs): 0.9625869 1.167079
## IC Z-Assintótico: 0.9638366 1.16583
```

```r
# g) Inversa Gaussiana
mu_ig_true <- 5; lambda_ig_true <- 2; n <- 100
x_ig <- rinvgauss(n, mean=mu_ig_true, shape=lambda_ig_true)

emv_mu_ig <- mean(x_ig)
emv_lambda_ig <- n / sum(1/x_ig - 1/emv_mu_ig)
# IC 1: Wald
var_mu_ig <- emv_mu_ig^3 / (n * emv_lambda_ig)
ic_wald_ig <- c(emv_mu_ig - z_score*sqrt(var_mu_ig),
```

```r
                emv_mu_ig + z_score*sqrt(var_mu_ig))
# IC 2: t-Student (Modificado)
t_score <- qt(1 - (1-alpha_conf)/2, df = n-1)
ic_t_ig <- c(emv_mu_ig - t_score*sqrt(var_mu_ig),
             emv_mu_ig + t_score*sqrt(var_mu_ig))

cat("g) Inversa Gaussiana ( Mu Real =", mu_ig_true, ")
    \nEMV:", emv_mu_ig, "\nIC Wald:", ic_wald_ig, "\nIC t-Student:", ic_t_ig)
```

```
## g) Inversa Gaussiana ( Mu Real = 5 )
##
## EMV: 4.641889
## IC Wald: 3.34686 5.936918
## IC t-Student: 3.330835 5.952943
```

```r
# h) Tweedie (p=1.5)
mu_tw_true <- 10; n <- 100
x_tw <- rtweedie(n, mu=mu_tw_true, phi=2, power=1.5)

emv_mu_tw <- mean(x_tw)
# Estimativa Phi (Momentos)
phi_hat <- var(x_tw) / (emv_mu_tw^1.5)

# IC 1: Assintótico GLM (Wald Z)
se_tw <- sqrt(phi_hat * emv_mu_tw^1.5 / n)
ic_glm_z <- c(emv_mu_tw - z_score*se_tw, emv_mu_tw + z_score*se_tw)

# IC 2: t-Student GLM
t_score_tw <- qt(1 - (1-alpha_conf)/2, df = n-1)
ic_glm_t <- c(emv_mu_tw - t_score_tw*se_tw, emv_mu_tw + t_score_tw*se_tw)

cat("h) Tweedie ( Mu Real =", mu_tw_true, ")
    \nEMV:", emv_mu_tw, "\nIC Wald (Z):", ic_glm_z, "\nIC t-Student:", ic_glm_t)
```

```
## h) Tweedie ( Mu Real = 10 )
##
## EMV: 10.25898
## IC Wald (Z): 8.57822 11.93973
## IC t-Student: 8.557422 11.96053
```