Lista 1

Modelagem com Apoio Computacional

Tailine J. S. Nonato

September 11, 2025

Carregamento dos dados:

```r
fatigue_df <- data.frame(
    work_MJ_m3 = c(
11.5,13.0,14.3,15.6,16.0,17.3,19.3,21.1,21.5,22.6,22.6,24.0,
24.0,24.6,25.2,25.5,26.3,27.9,28.3,28.4,28.6,30.9,31.9,34.5,
40.1,40.1,43.0,44.1,46.5,47.3,48.7,52.9,56.6,59.9,60.2,
60.3,60.5,62.1,62.8,66.5,67.0,67.1,67.9,68.8,75.4,100.5 ),
    life_cycles = c(
3280,5046,1563,4707,977,2834,2266,2208,1040,700,1583,482,
804,1093,1125,884,1300,852,580,1066,1114,386,745,736,
750,316,456,552,355,242,190,127,185,255,195,
283,212,327,373,125,187,135,245,137,200,190))


head(fatigue_df)
```

```
  work_MJ_m3 life_cycles
1       11.5        3280
2       13.0        5046
3       14.3        1563
4       15.6        4707
5       16.0         977
6       17.3        2834
```

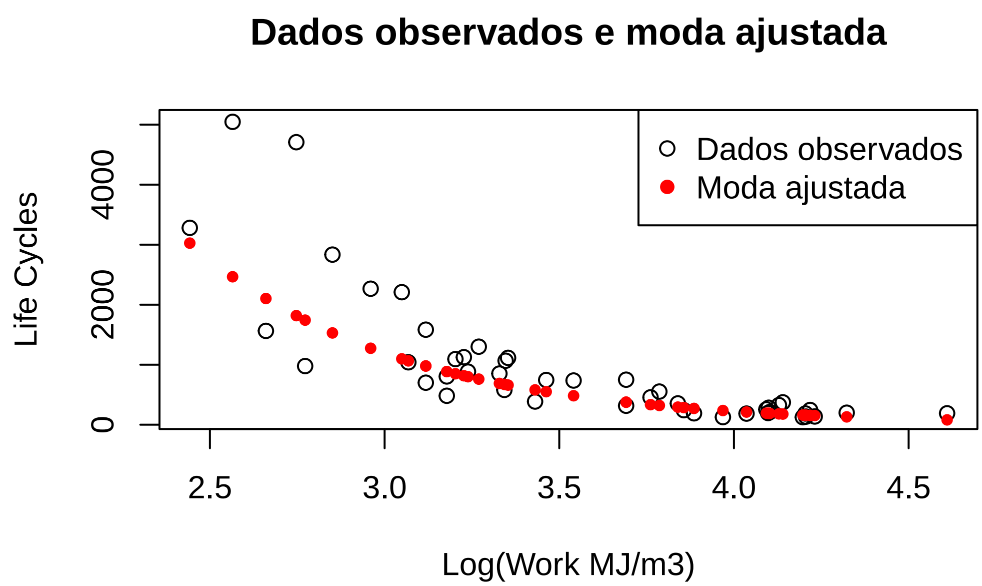
Função para ajuste do modelo de moda (log-BS):

```r
mle_mode_bs <- function(x, t) {
  x <- as.matrix(x)
  t <- as.matrix(t)

  fit_initial_log <- lm.fit(x, log(t))
  beta_initial_log_mode <- c(fit_initial_log$coef)

  k <- length(beta_initial_log_mode)
  n <- length(t)

  mu_initial_log <- x %*% beta_initial_log_mode
  alpha_initial_approx <- sqrt((4 / n) *
  sum((sinh((log(t) - mu_initial_log) / 2)) ^ 2))
  phi_initial <- alpha_initial_approx^2

  phi_initial <- max(0.01, min(0.99, phi_initial))

  thetaStar <- c(beta_initial_log_mode, phi_initial)

  loglik_mode <- function(par) {
    log_mode_Y <- x %*% par[1:k]
    phi_param <- par[k+1]
    if (phi_param <= 0 || phi_param >= 1) {
      return(NA)}

    alpha_orig <- sqrt(phi_param)
    beta_orig <- exp(log_mode_Y) / (1 - phi_param)
    terms <- log(t)

    l_i <- -log(alpha_orig) - 0.5 * log(2 * pi) - 0.5 * terms +
           log(sqrt(t / beta_orig) + sqrt(beta_orig / t)) -
           (1 / (2 * alpha_orig^2)) * (t / beta_orig + beta_orig / t - 2)

    if (any(!is.finite(l_i))) {
      return(.Machine$double.xmax)}

    return(-sum(l_i))}

  est <- optim(
    par = thetaStar,
    fn = loglik_mode,
    method = "BFGS",
```

```
    hessian = TRUE,
    control = list(maxit = 2000, reltol = 1e-12))

  if (est$conv != 0) {
    warning("FUNCTION DID NOT CONVERGE!")
  }

  coef <- (est$par)[1:k]
  phi_est <- est$par[k + 1]

  mode_hat_log <- x %*% coef
  mode_hat <- exp(mode_hat_log)

  SHess = solve(est$hessian)
  SE = sqrt(diag(SHess))

  tval = est$par / SE
  matcoef = cbind(est$par, SE, tval, 2 * (1 - pnorm(abs(tval))))

  AIC <- 2 * est$value + 2 * (k + 1)
  BIC <- 2 * est$value + (k + 1) * log(n)

  result <- list(
    phiHat = phi_est,
    betaHat_log_mode = coef,
    modeHat = mode_hat,
    AIC = AIC,
    BIC = BIC,
    matcoef = matcoef
  )

  return(result)
}
```

Estimativas do modelo com base na moda:

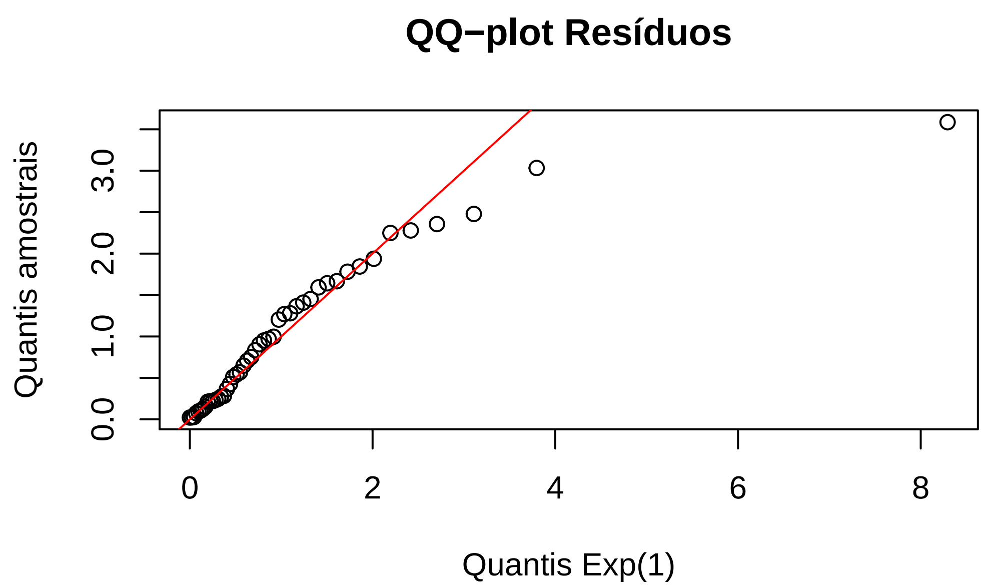
```
                                        SE       tval
(Intercept)                12.095313 0.39169573  30.879358 0.00000e+00
log(fatigue_df$work_MJ_m3) -1.670763 0.10844480 -15.406574 0.00000e+00
                            0.168396 0.03510972   4.796278 1.61641e-06
```



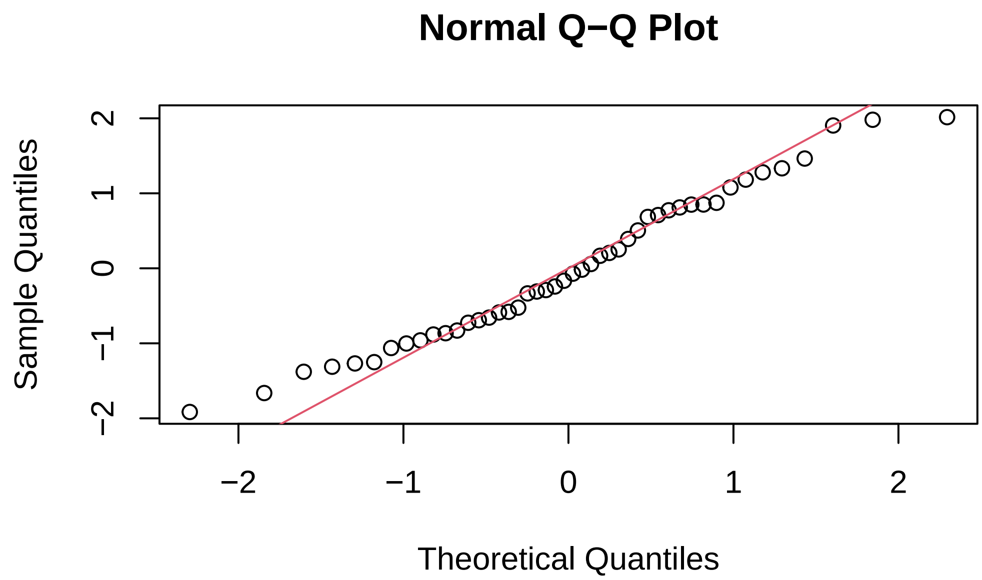
Análise do ajuste do modelo com base na moda:

```
S_bs_mode <- function(y, mu, phi) {
  alpha <- sqrt(phi)
  beta  <- mu / (1 - phi)
  xi <- (sqrt(y/beta) - sqrt(beta/y)) / alpha
  S <- 1 - pnorm(xi)
  return(S)
}

residuos_bs_mode <- function(y, mu_hat, phi_hat) {
  S <- S_bs_mode(y, mu_hat, phi_hat)
  r_cox <- -log(S)
  r_quant <- qnorm(S)
  list(coxsnell = r_cox, quantile = r_quant)
}


res_mode <- residuos_bs_mode(
  y = fatigue_df$life_cycles,
  mu_hat = fit_mode_bs$modeHat,
  phi_hat = fit_mode_bs$phiHat
)
```

```
# QQ plot
a <- ppoints(2000)
QGG <- qexp(a)
qqplot(QGG, res_mode$coxsnell,
       xlab = "Quantis Exp(1)", ylab = "Quantis amostrais",
       main = "QQ-plot Resíduos")
abline(0,1,col="red")
```




```
# QQ plot Quantílicos
qqnorm(res_mode$quantile); qqline(res_mode$quantile, col=2)
```



Ambos gráficos são bem similares aos gráficos para o modelo baseado na média.

Comparação com o modelo com base na média:

```
                                          se.coef       tval  
(Intercept)                 12.2797340 0.3893978  31.535187 0
log(fatigue_df$work_MJ_m3) -1.6707690 0.1084439 -15.406766 0
                             0.4103574 0.0427819   9.591846 0
```