Lista 1

Modelagem com Apoio Computacional

Tailine J. S. Nonato

September 23, 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
```

Ajuste do modelo com base na moda

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

```r
fit <- lm.fit(x, log(t))
beta <- c(fit$coef)

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

mu_init <- x %*% beta
alpha <- sqrt((4/n)*sum((sinh((log(t) - mu_init)/2))^2))

phi_init <- alpha^2 #chute inicial

thetaStar <- c(beta, phi_init)

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

  alpha_art <- sqrt(phi_param) #proposta do artifo
  beta_art <- exp(log_moda_Y) / (1 - phi_param) #proposta do artigo

  #pdf da distribuição log-BS
  z <- (sqrt(t/beta_art) - sqrt(beta_art/t))/alpha_art
  pdf_bs <- (1/(alpha_art*sqrt(t)))*(sqrt(t/beta_art) +
  sqrt(beta_art/t))*dnorm(z)
  pdf_bs[pdf_bs <= 0] <- .Machine$double.xmin
  #substitui valor menor ou igual a 0 por um número muito pequeno

  #log-verossimilhança
  l_i <- log(pdf_bs)
  if (any(!is.finite(l_i))) {
    return(.Machine$double.xmax)}
    #retorna um valor muito grande se algum log for infinito

  #verossimilhança total (loglik-plus)
  return(-sum(l_i))}

est <- optim(
  par = thetaStar,
  fn = loglik_moda,
  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]

  moda_hat_log <- x %*% coef
  moda_hat <- exp(moda_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_moda = coef,
    modaHat = moda_hat,
    AIC = AIC,
    BIC = BIC,
    matcoef = matcoef)

  return(result)}
```

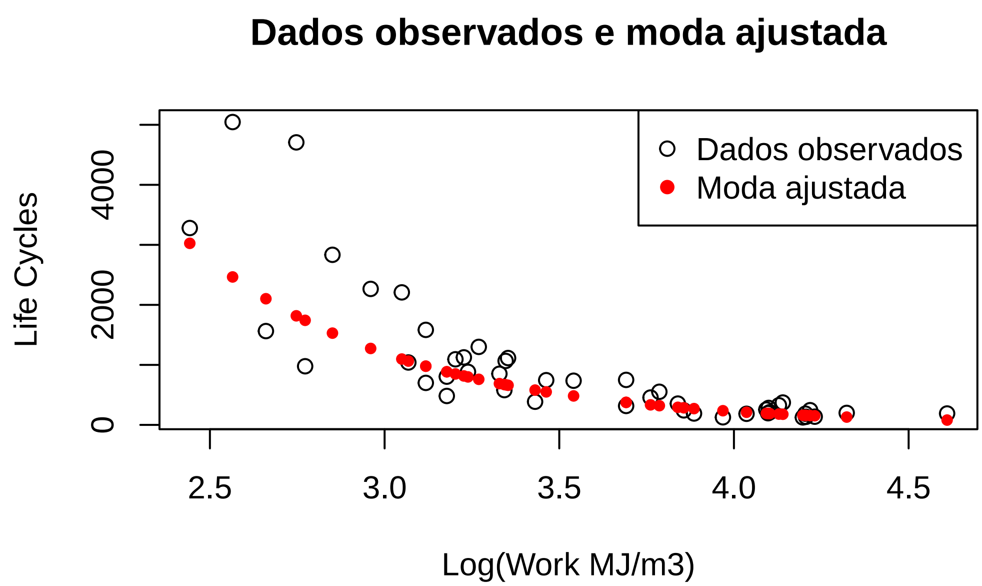
Estimativas do modelo

```
Zx <- model.matrix(~ log(fatigue_df$work_MJ_m3))
Zy <- fatigue_df$life_cycles

fit_moda <- mle_moda_bs(x = Zx , t = Zy)
fit_moda$matcoef
```

```
                                          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
```

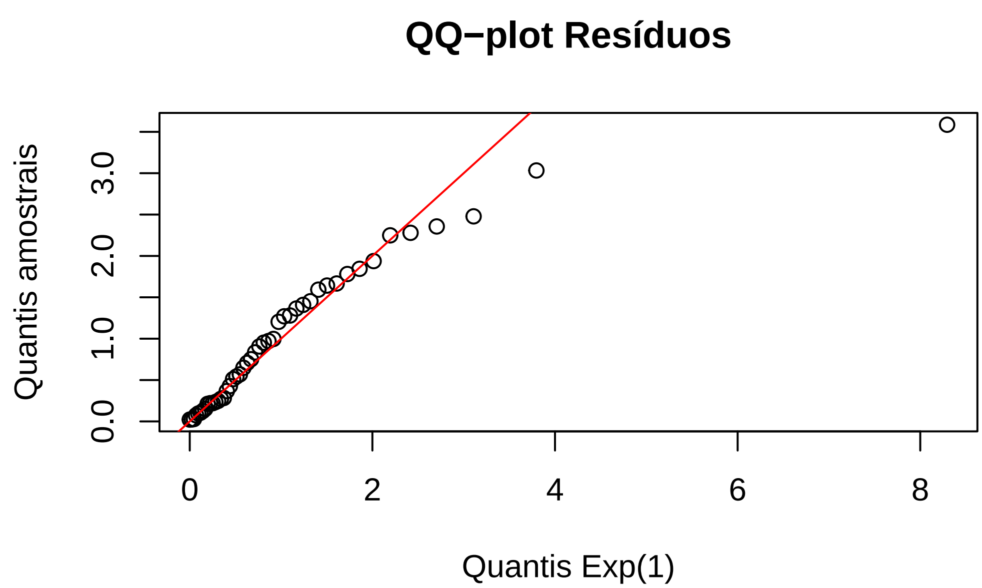
```
plot(log(fatigue_df$work_MJ_m3), Zy,
     main = "Dados observados e moda ajustada",
     xlab = "Log(Work MJ/m3)", ylab = "Life Cycles")
points(log(fatigue_df$work_MJ_m3), fit_moda$modaHat,
       col = "red", pch = 16, cex = 0.8)
legend("topright",
    legend = c("Dados observados", "Moda ajustada"),
    col = c("black", "red"), pch = c(1, 16))
```



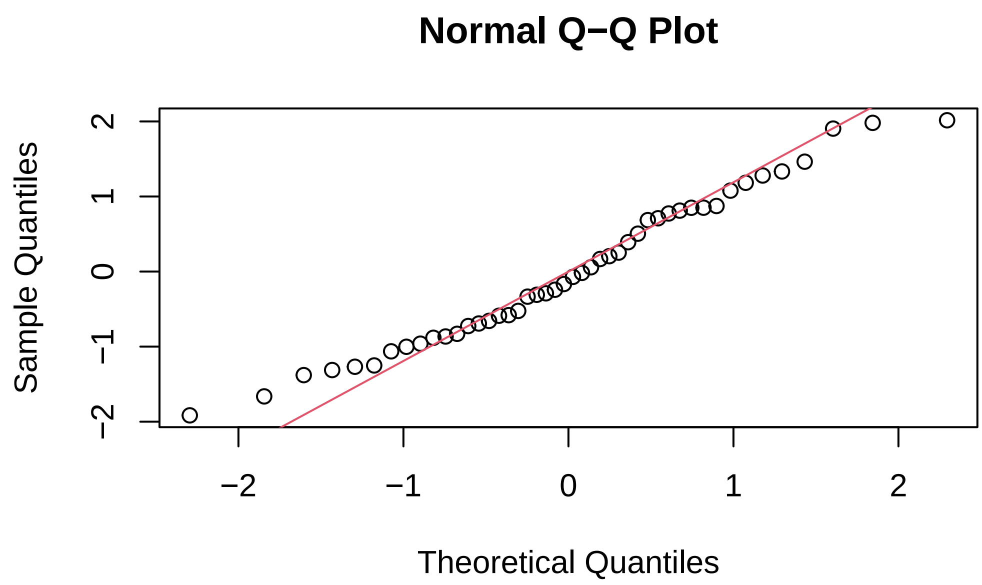
Verificação de ajuste do modelo

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

```r
  return(S)}

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

res_moda <- residuos_bs_moda(
  y = fatigue_df$life_cycles,
  mu_hat = fit_moda$modaHat,
  phi_hat = fit_moda$phiHat)

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



```r
# QQ plot quantílicos
qqnorm(res_moda$quantile); qqline(res_moda$quantile, col=2)
```



Os dois gráficos são bem similares aos gráficos para o modelo baseado na média. Indicando também adequação do modelo com base na moda.

Comparação dos modelos

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
```

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
```