Lista 2

Modelagem com Apoio Computacional

Tailine J. S. Nonato

November 25, 2025

Função do modelo de regressão Weibull quantílico

```
if (!require("pacman")) install.packages("pacman")
```

```
Loading required package: pacman
```

```
pacman::p_load(miscTools)

weibull_q_logpdf <- function(y, mu, k, q) {
  cq <- (-log(1 - q))^(1 / k)
  lambda <- mu / cq
  if (any(lambda <= 0)) stop("lambda <= 0 encontrado.")
  if (any(y <= 0))      stop("y <= 0 encontrado.")
  log_pdf <- log(k) - log(lambda) +
    (k - 1) * (log(y) - log(lambda)) -
    (y / lambda)^k
  as.vector(log_pdf)
}

weibull_q <- function(t, mu, k, q, log = FALSE) {
  lp <- weibull_q_logpdf(t, mu, k, q)
  if (log) return(lp)
  exp(lp)
}

mle_weibull_q <- function(x, t, k, q, print = TRUE) {
  x <- as.matrix(x)
```

```r
y <- as.vector(t)
n <- length(y)
p <- ncol(x)

loglik <- function(par) {
  beta <- par[1:p]
  eta <- as.vector(x %*% beta)
  mu  <- exp(eta)
  -sum(weibull_q_logpdf(y, mu, k, q))
}

theta0 <- rep(0, p)
est <- optim(theta0, loglik, method = "BFGS", hessian = TRUE,
             control = list(reltol = 1e-9, maxit = 1000))

beta_hat <- as.vector(est$par[1:p])
names(beta_hat) <- colnames(x)

vcov_beta <- solve(est$hessian)
se <- sqrt(diag(vcov_beta))

tb_beta <- miscTools::coefTable(beta_hat, se, df = (n - p))

etaHat <- as.vector(x %*% beta_hat)
muHat  <- exp(etaHat)
cq     <- (-log(1 - q))^(1 / k)
lambdaHat <- muHat / cq

AIC <- 2 * loglik(est$par) + 2 * p
BIC <- 2 * loglik(est$par) + log(n) * p

if (print) {
  cat("\n")
  cat("---------------------------------------------------------------\n")
  cat("         Weibull quantile regression model                     \n")
  cat("---------------------------------------------------------------\n")
  cat("Maximum Likelihood estimation \n")
  cat("Number of observations:", n, "\n")
  cat("Shape k (fixed):", k, " | Quantile q:", q, "\n")
  cat("---------------------------------------------------------------\n")
  cat("Beta - Coefficients (for log(Q_q)):\n")
  printCoefmat(tb_beta, signif.stars = TRUE, signif.legend = TRUE, digits = 4)
```

```
    cat("-----------------------------------------------------------------\n")
    cat("AIC:", round(AIC, 2),
        " BIC:", round(BIC, 2), "\n")
  }

  out <- list(
    par         = est$par,
    beta        = beta_hat,
    se          = se,
    vcov        = vcov_beta,
    muHat       = muHat,
    lambdaHat   = lambdaHat,
    value       = est$value,
    hessian     = est$hessian,
    convergence = est$convergence,
    table       = tb_beta,
    k           = k,
    q           = q
  )
  class(out) <- "mle_weibull_q"
  out
}
```

Geração de NPAs

```
set.seed(455)

n  <- 100
k  <- 0.5
q  <- 0.5
beta <- c(0.5, 1.0)

x1 <- rnorm(n)
X  <- cbind(1, x1)
colnames(X) <- c("(Intercept)", "x1")

eta <- as.vector(X %*% beta)
Q  <- exp(eta)
```

```r
lambda <- Q*((-log(1-q))^(-1/k))

y <- rweibull(n, shape = k, scale = lambda)
```

Estimação do modelo

```r
fit_weibull <- mle_weibull_q(x = X, t = y, k = k, q = q, print = TRUE)
```

```
------------------------------------------------------------
         Weibull quantile regression model
------------------------------------------------------------
Maximum Likelihood estimation
Number of observations: 100
Shape k (fixed): 0.5  | Quantile q: 0.5
------------------------------------------------------------
Beta - Coefficients (for log(Q_q)):
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.5298     0.2031   2.608   0.0105 *
x1            1.0185     0.1992   5.113 1.57e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
------------------------------------------------------------
AIC: 516.16  BIC: 521.37
```

```r
# Estimated beta coefficients:
fit_weibull$beta
```

```
(Intercept)          x1
  0.5297608   1.0185185
```

```r
# True beta coefficients:
beta
```

```
[1] 0.5 1.0
```