

## Resultados para o trabalho 6

```
## Trabalho 6
## Modelo t de Student

# Função densidade
dtv <- function(y, mu = 0, sigma2 = 1, nu = 50, log = FALSE) {
  cnu <- gamma(0.5 * (nu + 1)) / (sqrt((nu - 2) * pi) * gamma(0.5 * nu))
  d2 <- (y - mu)^2 / ((nu - 2) * sigma2)
  fy <- cnu / (sqrt(sigma2) * (1 + d2)^(0.5 * (nu + 1)))
  if (log) fy <- log(fy)
  fy
}

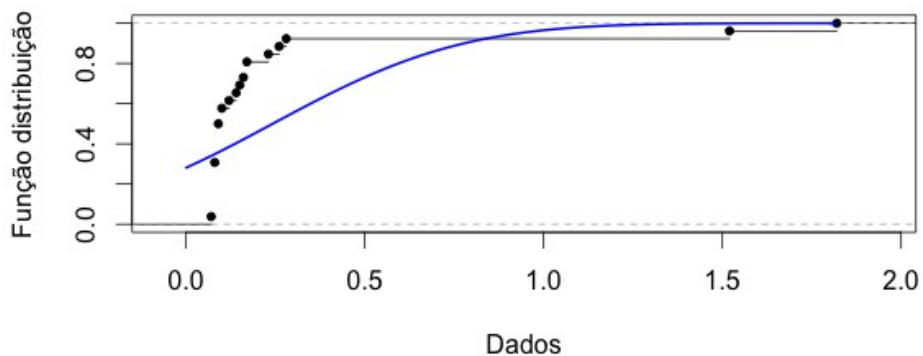
# Tolerância
eps <- 1e-6

# Dados (y)
dados <- scan()
0.17 0.15 0.08 0.08 1.52 0.09 0.08 0.28 0.08 0.09 1.82 0.09 0.08
0.26 0.09 0.09 0.07 0.10 0.08 0.14 0.08 0.16 0.12 0.17 0.23 0.10

n <- length(dados)
cat("\n n =", n)

      n = 26

# Modelo normal (EMV)
mucn <- mean(dados)
sigma2cn <- var(dados) * (n - 1) / n
plot(ecdf(dados), main = "", pch = 20, xlab = "Dados",
     ylab = "Função distribuição")
curve(pnorm(x, mucn, sqrt(sigma2cn)), add = TRUE, col = "blue",
      xlim = c(0, max(dados)), lwd = 2)
```



Nota 1. Comente sobre o ajuste com a distribuição normal.

```

# Estimação modelo t
vnu <- c(2.1, 2.5, 3:20) # Graus de liberdade
emvt <- matrix(NA, length(vnu), 2) # EMV modelo t
logver <- c()
for (i in 1:length(vnu)) {

  # Estimativas iniciais (EMV modelo normal)
  mu0 <- mucn
  sigma20 <- sigma2cn

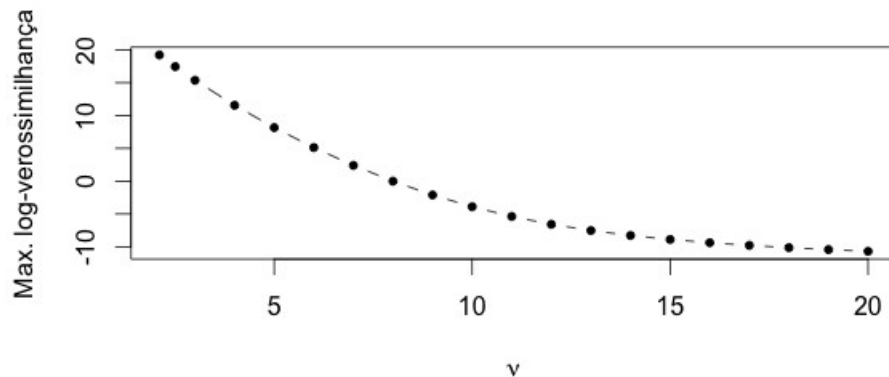
  iter <- 0
  dif <- 1
  while (dif > eps) {
    iter <- iter + 1

    # Passo E
    z <- (vnu[i] + 1) / (vnu[i] - 2 + (dados - mu0)^2 / sigma20)

    # Passo M
    muc <- weighted.mean(dados, w = z)
    sigma2c <- mean(z * (dados - muc)^2)

    # Critério de parada
    dif <- max(abs((muc - mu0) / mu0), abs((sigma2c - sigma20) /
      sigma20))
    mu0 <- muc
    sigma20 <- sigma2c
  }
  emvt[i,] <- c(muc, sigma2c)
  logver[i] <- sum(dtv(dados, mu = muc, sigma2 = sigma2c, nu = vnu[i],
    log = TRUE))
  cat("\n nu:", vnu[i], "\n Iterações:", iter, "mu, sigma2",
    c(muc, sigma2c))
}
plot(vnu, logver, type = "b", xlab = expression(nu),
  ylab = "Max. log-verossimilhança", pch = 20)

```



```
# EMV modelo t
imax <- which.max(logver)
cat("\n EMV modelo t (mu, sigma2, nu):", c(emvt[imax,], vnu[imax]))
```

EMV modelo t (mu, sigma2, nu): 0.1019012 0.03355534 2.1

```
# Pesos
plot(z, type = "h", xlab = "Observação", ylab = "Peso")
abline(h = 1, lty = 2, col = "blue", lwd = 2)
```

