

# plotZ.R

fulvio

2020-10-01

Gráfico da probabilidade de pertencer a uma área da curva Normal @aliases plotZ

@param x valor a comparar com a média @param mu média @param dp desvio-padrão @param p probabilidade  
@param z score-z @param cor cor do preenchimento da área sob a curva @param main título @param sub  
subtítulo @param sentido “menor” (padrão) calcula e desenha a probabilidade de um valor menor ou igual  
a x, p ou z; “maior” calcula e desenha a probabilidade de um valor maior que x, p ou z @param cex.main  
tamanho da fonte do título @param cex.sub tamanho da fonte do subtítulo @param cex.axis tamanho da  
fonte do eixo @param ... Permite o uso de outros parâmetros gráficos (veja o uso de **density**, nos exemplos)

@examples plotZ(p = .975) plotZ(p = .025) plotZ(z = 1.96, density = 20) plotZ(z = -1.96, density = 40)  
plotZ(x = 10, mu = 4.7, dp = 2.7, cor = “yellow”) plotZ(x = 10, mu = 4.7, dp = 2.7, sentido = “maior”)  
plotZ(p = .975, sentido = “maior”) plotZ(p = .025, sentido = “maior”) plotZ(z = 1.96, sentido = “maior”)  
plotZ(z = -1.96, sentido = “maior”)

@export

```
plotZ <- function(x = NULL, mu = 0, dp = 1, p = NULL, z = NULL,
                 cor = 2, main = NULL, sub = NULL,
                 sentido = 'menor',
                 cex.main = 2, cex.sub = 1.5, cex.axis = 1.3,
                 ...) {
  # if(!is.null(p) & (p<0 | p>1)) {
  #   cat("ERRO:\tA probabilidade deve estar entre 0 e 1.")
  #   return()
  # }
  titulo.menor <- bquote(P(X <= .(x),
                        italic(N(bar(x))==(formatC(mu, decimal.mark = ",")),
                        s==(formatC(dp, decimal.mark = ","))))
  titulo.maior <- bquote(P(X > .(x),
                        italic(N(bar(x))==(formatC(mu, decimal.mark = ",")),
                        s==(formatC(dp, decimal.mark = ","))))

  if(!is.null(x)) {
    zp <- z <- (x - mu)/dp
    if(is.null(main)) {
      if(sentido == "menor") {
        main <- titulo.menor
      } else if(sentido == "maior") {
        main <- titulo.maior
      }
    }
  }
}
```

```

if(!is.null(p)) {
  zp = qnorm(p)
} else if(!is.null(z)) {
  zp = z
  if(sentido == "menor") {
    main <- titulo.menor
  } else if(sentido == "maior") {
    main <- titulo.maior
  }
  # if(is.null(x)) main = ""
}

if(is.null(main)) {
  main <- expression(italic(N(mu==0, sigma==1)))
}

if(is.null(sub)) {
  if(sentido == 'menor') {
    if(!is.null(p)) {
      prob <- bquote(P(X <= .(formatC(qnorm(p), 3, format = "f", decimal.mark = ","))) == .(p))
      # prob <- bquote(P(X <= .(p)) == .(round(qnorm(p), 3)))
    } else if(!is.null(z)) {
      prob <- bquote(P(Z <=
        .(formatC(z, 2, format = "f", decimal.mark = ","))) ==
        .(formatC(pnorm(z), 3, format = "fg", decimal.mark = ",")))
    }
  } else if (sentido == 'maior') {
    if(!is.null(p)) {
      prob <- bquote(P(X > .(formatC(1-qnorm(p), 3, format = "f", decimal.mark = ","))) == .(p))
    } else if(!is.null(z)) {
      prob <- bquote(P(Z > .(formatC(z, 2, format = "f", decimal.mark = ","))) == .(formatC(1-pno
    }
  }
} else
  prob <- sub
d <- rnorm(1000000)
plot(density(d), lwd=2, xlim = c(-3.5,3.5),
  axes = F, cex.lab = 1.3,
  xlab = expression(italic(Z)), ylab = "",
  main = main,
  cex.main = cex.main)
axis(1, -4:4, cex.axis = cex.axis)
mtext(prob, 3, -.25, cex = cex.sub)
# polygon(c(-4, seq(-4,4, .1), 4), c(0, dnorm(seq(-4, 4, .1)), 0), col = 0)
if(sentido == 'menor') {
  x.val <- c(-4, seq(-4, zp, 0.01), zp)
  y.val <- c(0, dnorm(seq(-4, zp, 0.01)), 0)
} else if (sentido == 'maior') {
  x.val <- c(zp, seq(zp, 4, 0.01), 4)
  y.val <- c(0, dnorm(seq(zp, 4, 0.01)), 0)
}
polygon(x.val, y.val, col = cor, border = NA, ...)
}

```