Distribuições contínuas

2024

Algumas distribuições contínuas são apresentadas. A linguagem R é utilizada.

Várias distribuições de probabilidade contínuas estão implementadas no pacote `stats`, que é um dos pacotes básicos da linguagem R. Em geral, se X denota uma variável aleatória, para uma distribuição qualquer identificada por `nome`, existem quatro funções, brevemente descritas abaixo.

1. `dnome`: função densidade de probabilidade $f(x)$.
2. `pnome`: função distribuição acumulada $F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du$.
3. `qnome`: função quantil.
4. `rnome`: geração de amostras da distribuição.

```
# Separador decimal nos resultados: ","
options(OutDec = ",")
```

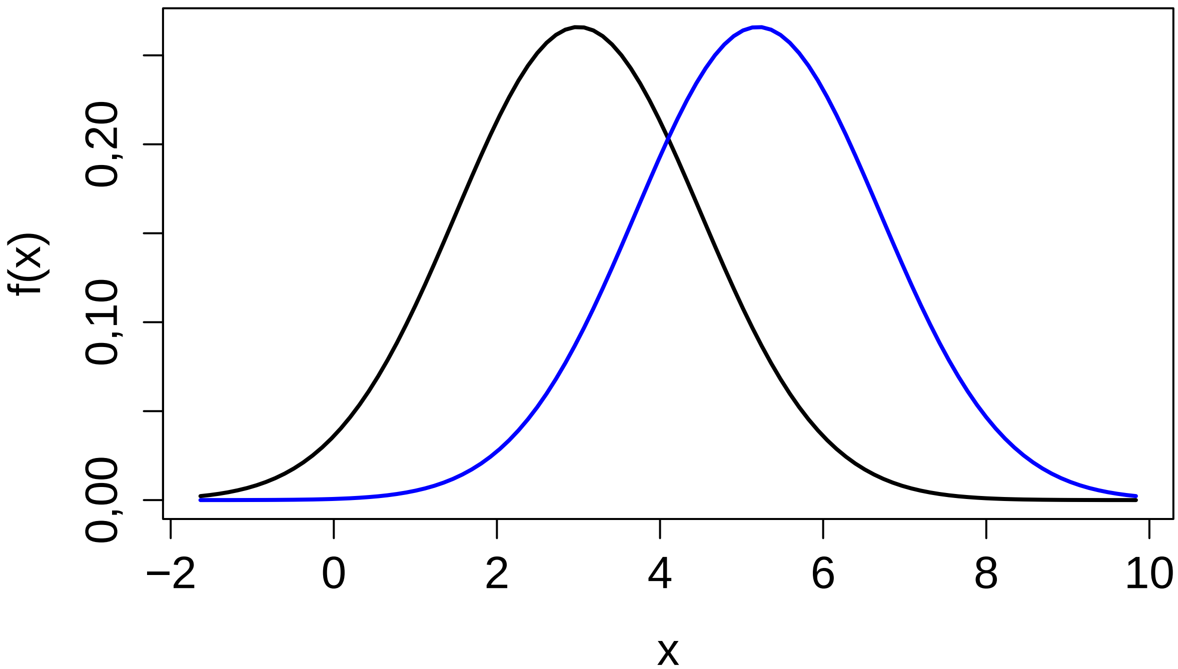
1. Normal

Sintaxe básica: `dnorm(x, mu, sigma)`.

Vale notar que o segundo parâmetro é o desvio padrão σ.

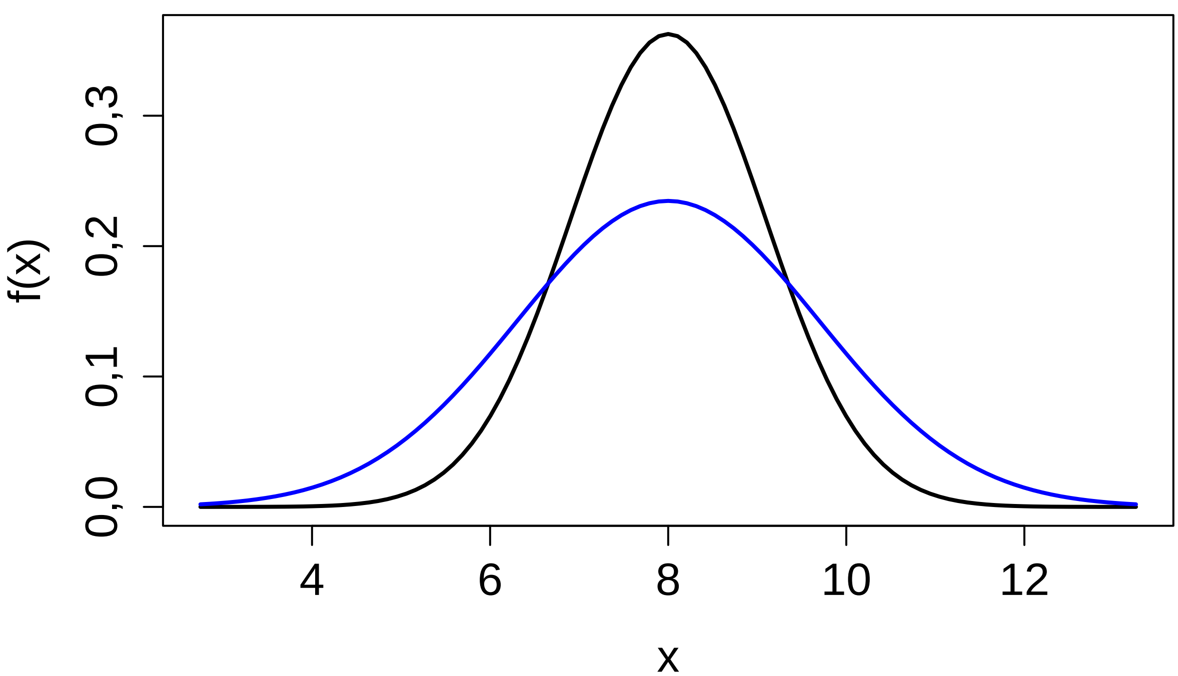
```
# Diferentes médias
mean1 <- 3
mean2 <- 5.2
sd1 <- 1.5

curve(dnorm(x, mean = mean1, sd = sd1), xlab = "x", ylab = "f(x)",
   from = qnorm(0.001, mean1, sd1), to = qnorm(0.999, mean2, sd1),
   lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dnorm(x, mean = mean2, sd = sd1), add = TRUE, col = "blue",
      lwd = 2)
```



Nota 1. Procure entender todos os argumentos do comando acima.

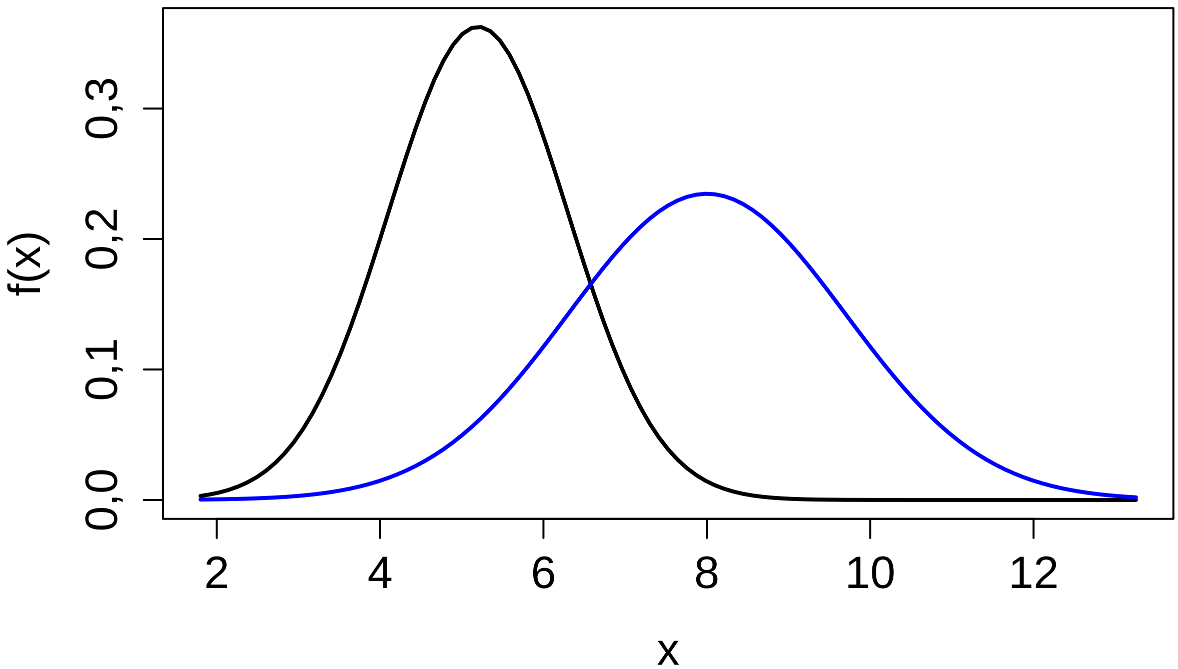
```
# Diferentes desvios padrão
mean1 <- 8
sd1 <- 1.1
sd2 <- 1.7
curve(dnorm(x, mean = mean1, sd = sd1), xlab = "x", ylab = "f(x)",
      from = qnorm(0.001, mean1, sd2), to = qnorm(0.999, mean1, sd2),
      lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dnorm(x, mean = mean1, sd = sd2), add = TRUE, col = "blue",
      lwd = 2)
```



```r
# Diferentes médias e desvios padrão
mean1 <- 5.2
mean2 <- 8
sd1 <- 1.1
sd2 <- 1.7

xmin <- min(qnorm(0.001, mean1, sd1), qnorm(0.001, mean2, sd2))
xmax <- max(qnorm(0.999, mean1, sd1), qnorm(0.999, mean2, sd2))

curve(dnorm(x, mean = mean1, sd = sd1), xlab = "x", ylab = "f(x)",
      from = xmin, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dnorm(x, mean = mean2, sd = sd2), add = TRUE, col = "blue",
      lwd = 2)
```



Calculamos $P(\mu - j\sigma \leq X \leq \mu + j\sigma \leq)$, para $j = 1, 2, 3$, notando que

$$P(\mu - j\sigma \leq \; X \; \leq \mu + j\sigma) = P(-j \leq \; Z \; \leq j),$$

em que

$$Z = \frac{X - \mu}{\sigma} \;\sim\; N(0, 1).$$

```r
for (j in 1:3) {
  cat("j =", j, ":", pnorm(j) - pnorm(-j), "\n")
}
```

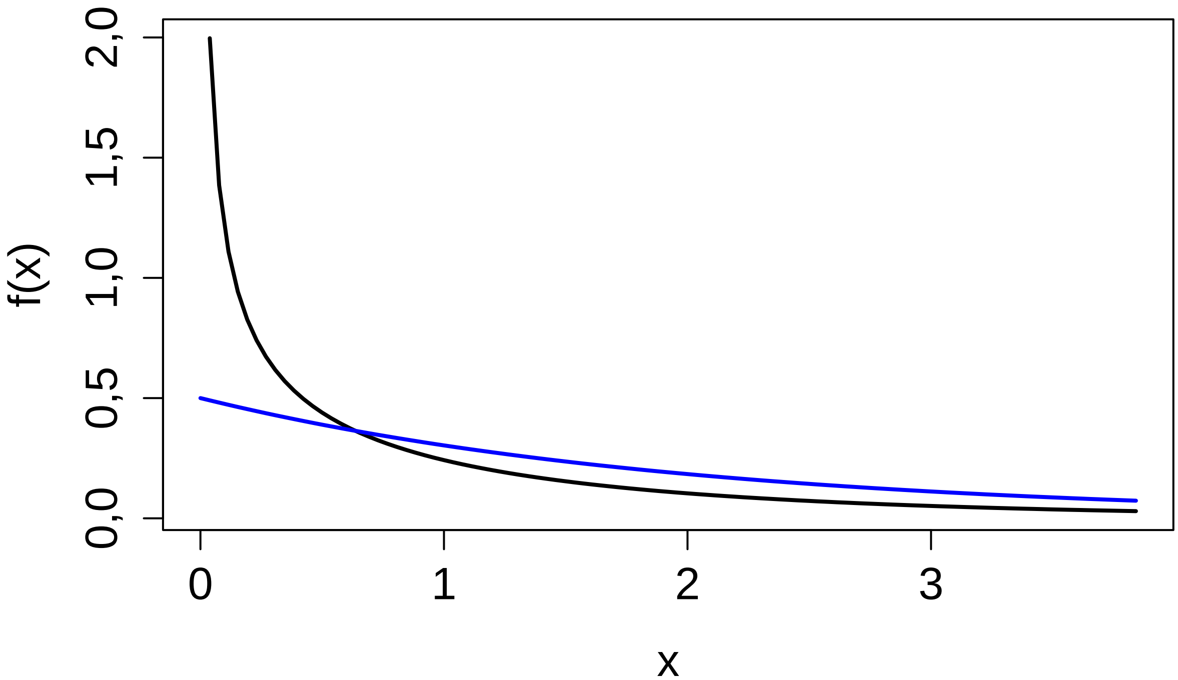
```
## j = 1 : 0,6826895
## j = 2 : 0,9544997
## j = 3 : 0,9973002
```

2. Qui-quadrado

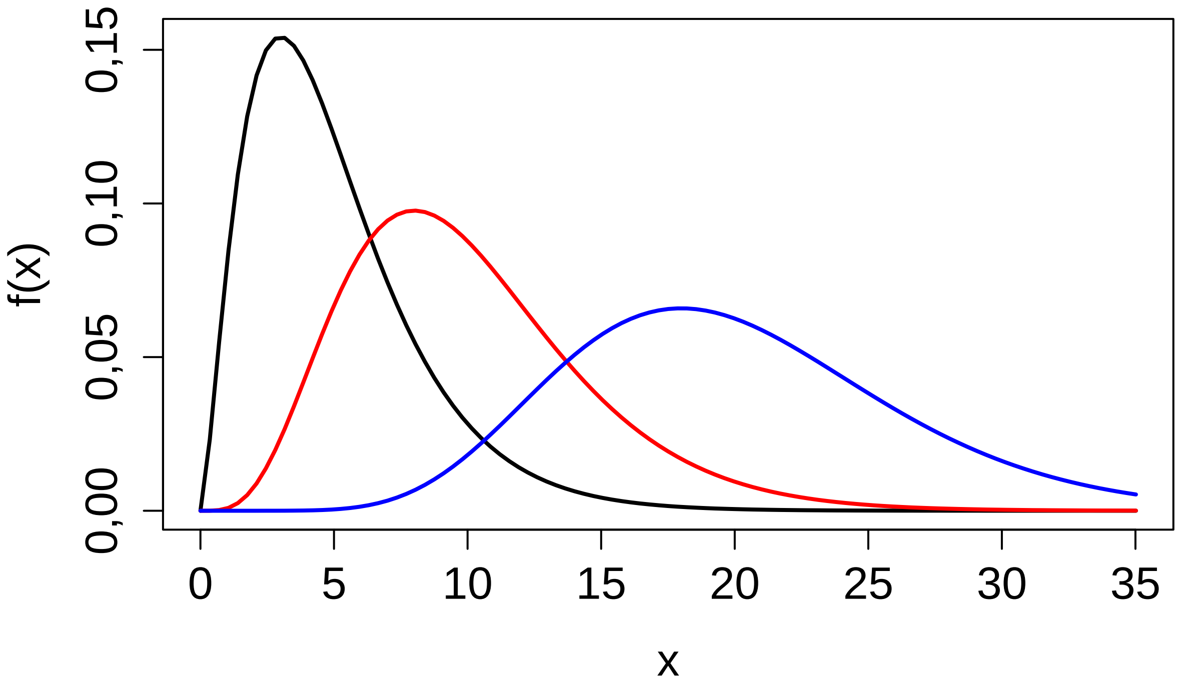
Sintaxe básica: `dchisq(x, gl)`,

em que `gl` representa os graus de liberdade k.

```
xmax <- qchisq(0.95, 1)
curve(dchisq(x, 1), xlab = "x", ylab = "f(x)",
      from = 0, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dchisq(x, 2), add = TRUE, col = "blue", lwd = 2)
```




```
xmax <- qchisq(0.98, 20)
curve(dchisq(x, 5), xlab = "x", ylab = "f(x)",
      from = 0, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dchisq(x, 10), add = TRUE, col = "red", lwd = 2)
curve(dchisq(x, 20), add = TRUE, col = "blue", lwd = 2)
```

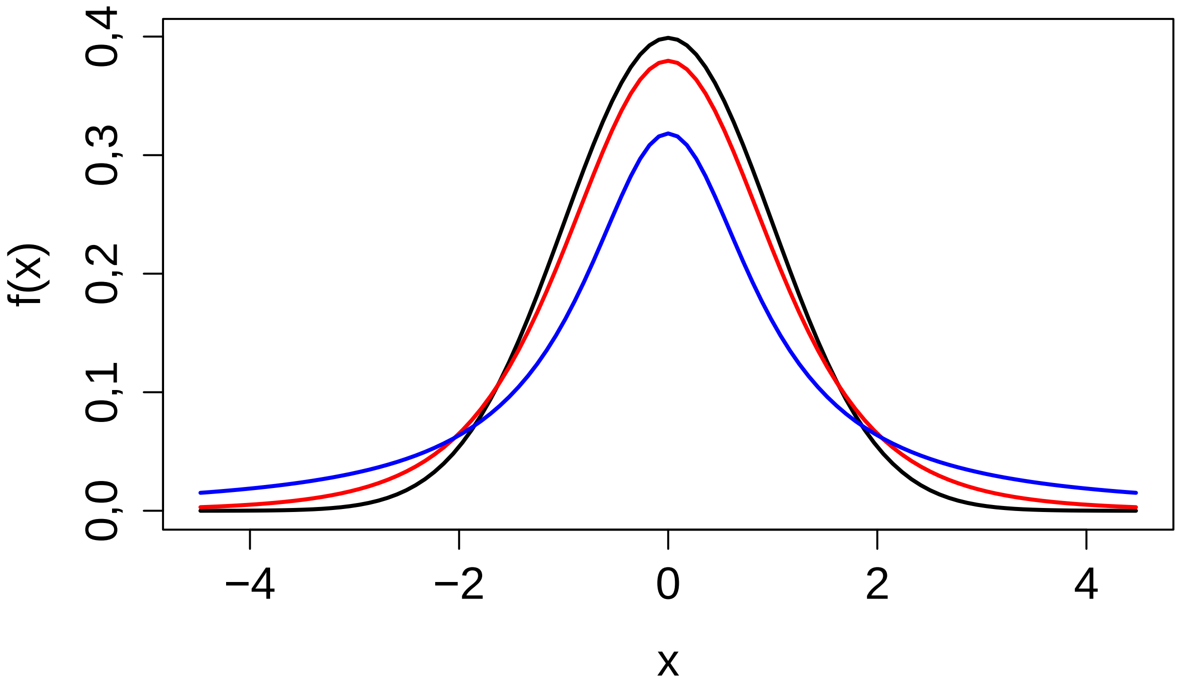


3. t de Student

Sintaxe básica: `dt(x, gl)`,

em que `gl` representa os graus de liberdade k. A distribuição Cauchy corresponde a $k = 1$. Quando $k \to \infty$, a distribuição t de Student tende à distribuição $N(0, 1)$.

```
xmax <- qt(0.93, 1)
curve(dnorm(x, mean = 0, sd = 1), xlab = "x", ylab = "f(x)",
      from = -xmax, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dt(x, 5), add = TRUE, col = "red", lwd = 2)
curve(dt(x, 1), add = TRUE, col = "blue", lwd = 2)
```

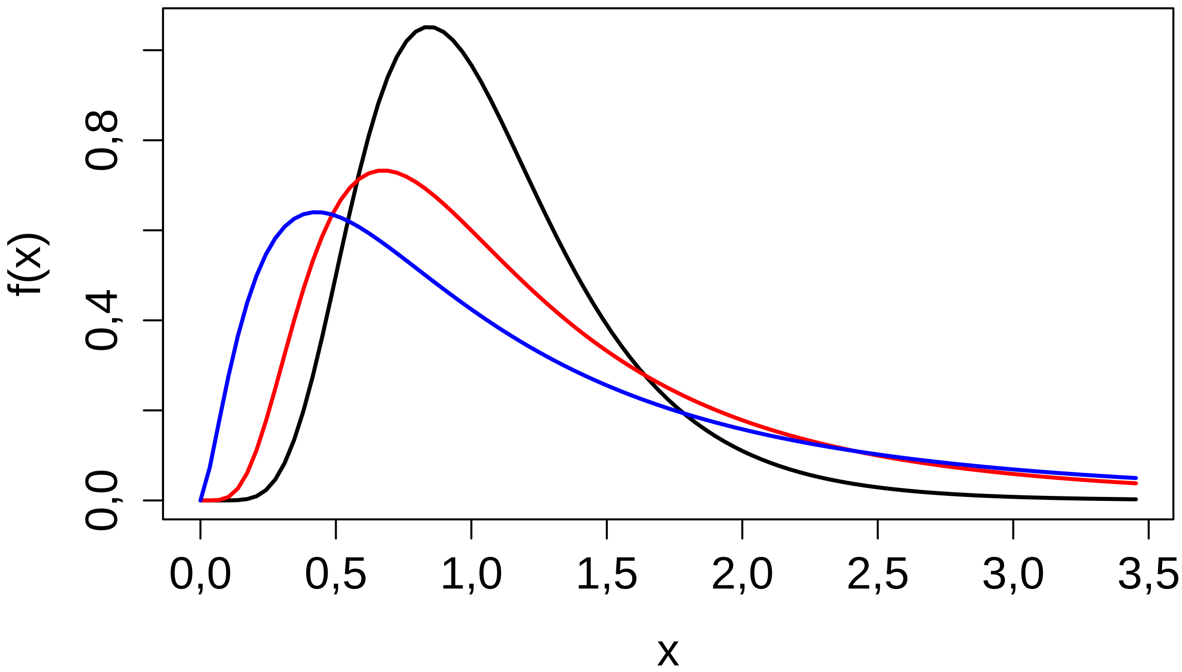


4. F de Snedecor-Fisher

Sintaxe básica: `df(x, gl1, gl2)`,

em que `gl1` e `gl2` representam os graus de liberdade do numerador e denominador k e l.

```
xmax <- qf(0.9, 5, 5)
curve(df(x, 20, 30), xlab = "x", ylab = "f(x)",
      from = 0, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(df(x, 15, 7), add = TRUE, col = "red", lwd = 2)
curve(df(x, 5, 5), add = TRUE, col = "blue", lwd = 2)
```

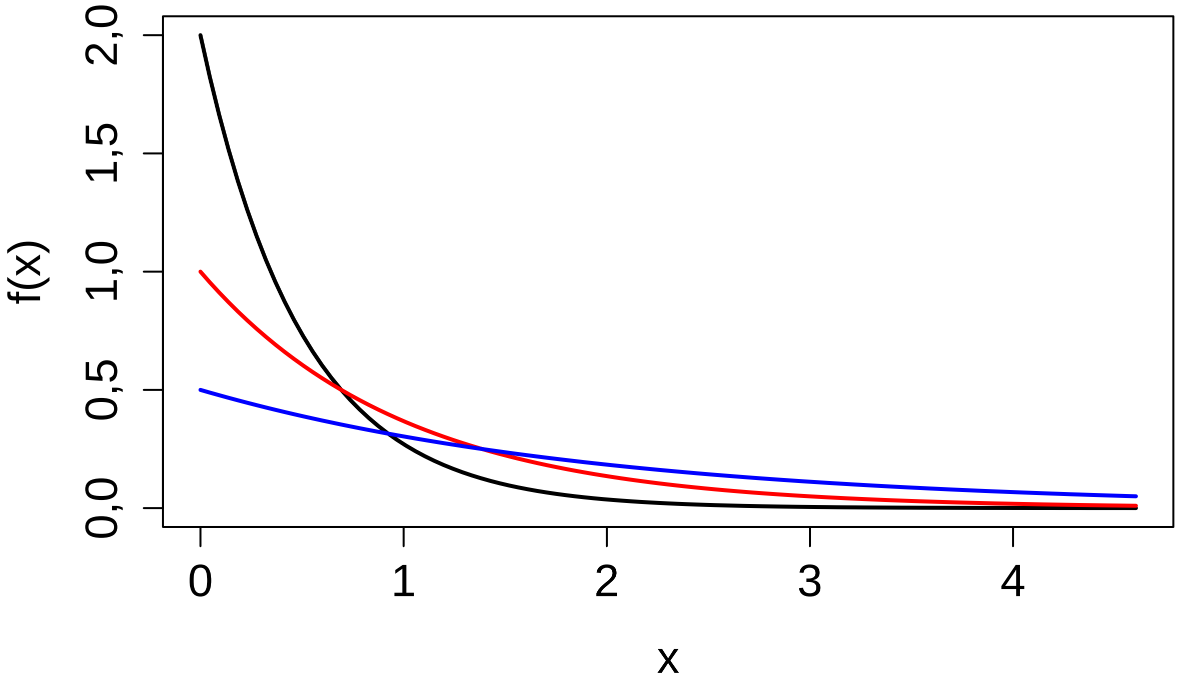


5. Exponencial

Sintaxe básica: `dexp(x, teta)`,

em que `teta` é o parâmetro de taxa, ou seja, $f(x) = \theta e^{-\theta x}$, se $x > 0$.

```
xmax <- qexp(0.9, 0.5)
curve(dexp(x, 2), xlab = "x", ylab = "f(x)",
      from = 0, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dexp(x, 1), add = TRUE, col = "red", lwd = 2)
curve(dexp(x, 0.5), add = TRUE, col = "blue", lwd = 2)
```



6. Gama

Sintaxe básica: `dgamma(x, alfa, teta)`,

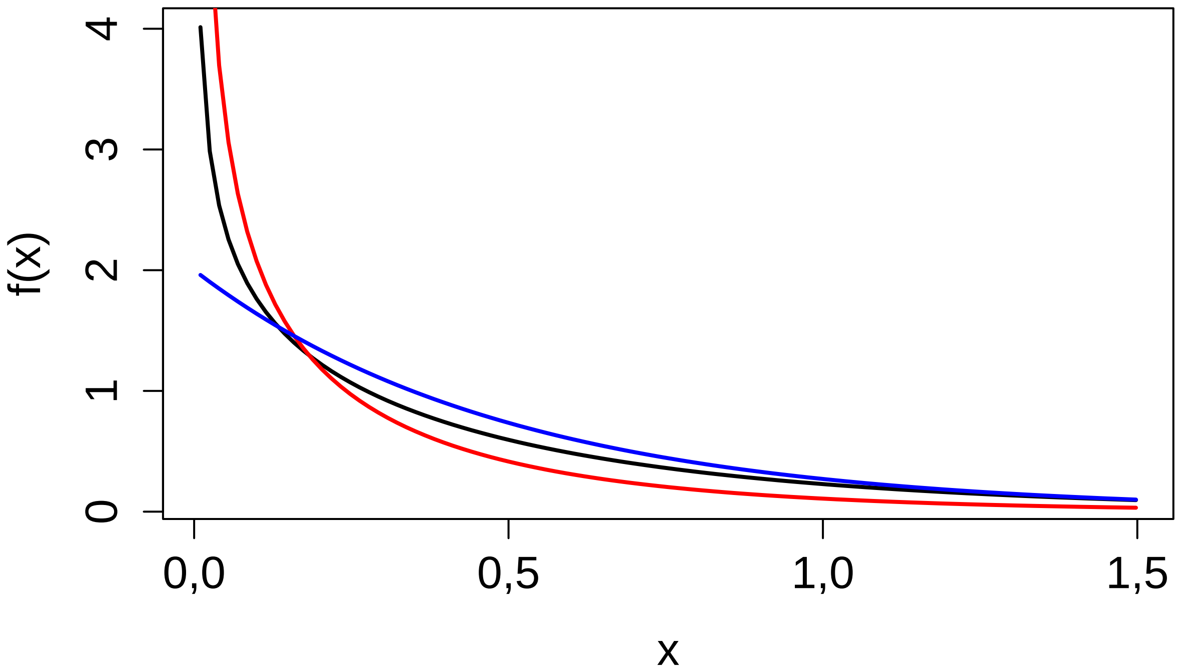
em que `alfa` e `teta` sào os parâmetros de forma e taxa, respectivamente, ou seja,

$$f(x) = \frac{\theta^{\alpha} x^{\alpha-1} e^{-\theta x}}{\Gamma(\alpha)}, \quad \text{se } x > 0.$$

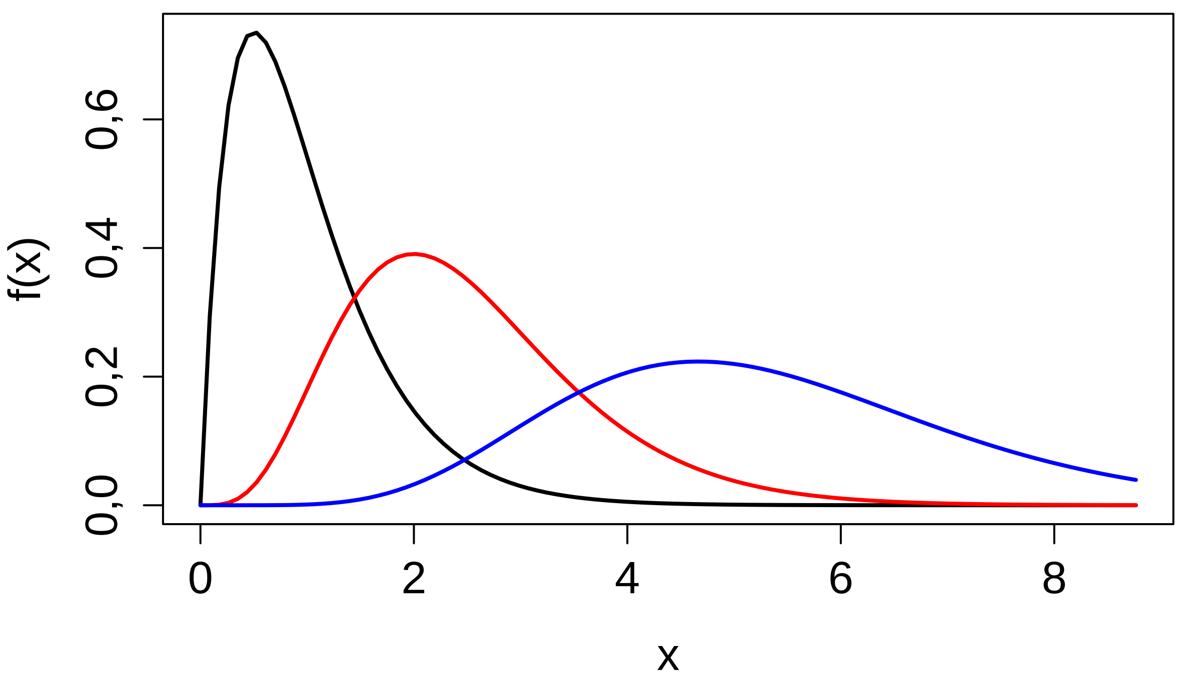
Casos particulares:

1. Se $\alpha = 1$, obtemos a distribuição exponencial com parâmetro de taxa θ.
2. Se $\alpha = k/2$ e $\theta = 1/2$, obtemos a distribuição qui-quadrado com k graus de liberdade.

```
xmax <- qgamma(0.95, 1, 2)
curve(dgamma(x, 0.7, 1.5), xlab = "x", ylab = "f(x)",
      from = 0.01, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dgamma(x, 0.5, 2), add = TRUE, col = "red", lwd = 2)
curve(dgamma(x, 1, 2), add = TRUE, col = "blue", lwd = 2)
```




```r
xmax <- qgamma(0.95, 8, 1.5)
curve(dgamma(x, 2, 2), xlab = "x", ylab = "f(x)",
      from = 0, to = xmax, lwd = 2, cex.axis = 1.4, cex.lab = 1.4)
curve(dgamma(x, 5, 2), add = TRUE, col = "red", lwd = 2)
curve(dgamma(x, 8, 1.5), add = TRUE, col = "blue", lwd = 2)
```



Nota 2. Refaça os exemplos em linguagem Python.

Nota 3. Diversas distribuições de probabilidade, tanto discretas quanto contínuas, estão disponíveis na linguagem R e estão descritas na página *CRAN Task View: Probability Distributions* (https://cran.r-project.org/web/views/Distributions.html).