

```
### Testes de normalidade
```

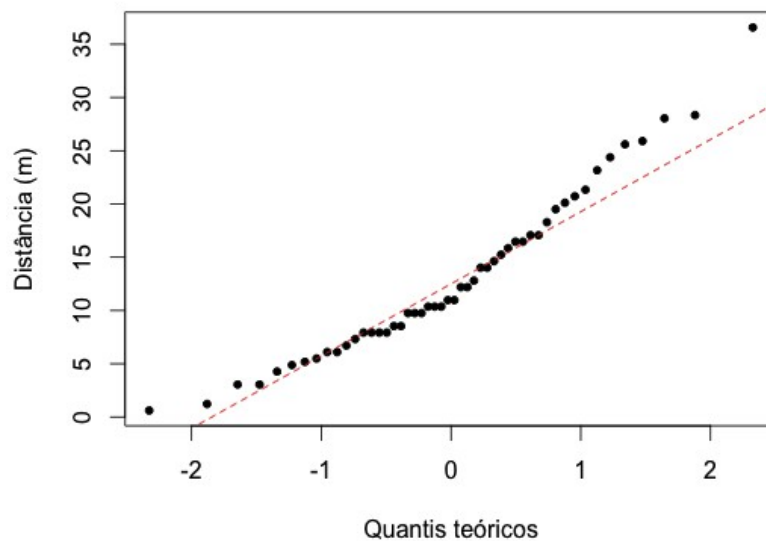
```
# Pacote com alguns testes  
library(nortest)
```

```
# Dados. Distância percorrida por carros até parar (em pés)  
# Conjunto de dados "cars"  
dados <- cars$dist  
dados <- dados * 12 * 2.54 / 100 # em m
```

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

```
## 1. Testes de normalidade
```

```
# Gráfico de probabilidade (QQ)  
qqnorm(dados, main = "", xlab = "Quantis teóricos", pch = 20,  
        ylab = "Distância (m)")  
qqline(dados, lty = 2, col = "red")
```



```
# Estimativas dos parâmetros  
xb <- mean(dados) # mu  
sx <- sd(dados)   # sigma  
cat("\n Média amostral =", xb, "\n Desvio padrão amostral =", sx)
```

```
# Testes  
(t1 <- ks.test(dados, "pnorm", xb, sx) ) # KS  
(t2 <- lillie.test(dados)) # Lilliefors
```

```

(t3 <- cvm.test(dados)) # Cramér-von Mises
(t4 <- shapiro.test(dados)) # Shapiro-Wilk
(t5 <- sf.test(dados)) # Shapiro-Francia
(t6 <- ad.test(dados)) # Anderson-Darling

# Tabela de resultados
testes <- c(t1$method, t2$method, t3$method, t4$method, t5$method,
            t6$method)
estt <- as.numeric(c(t1$statistic, t2$statistic, t3$statistic,
                    t4$statistic, t5$statistic, t6$statistic))
valorp <- c(t1$p.value, t2$p.value, t3$p.value, t4$p.value,
            t5$p.value, t6$p.value)
resultados <- cbind(estt, valorp)
rownames(resultados) <- testes
colnames(resultados) <- c("Estatística", "p")
print(resultados, digits = 4)

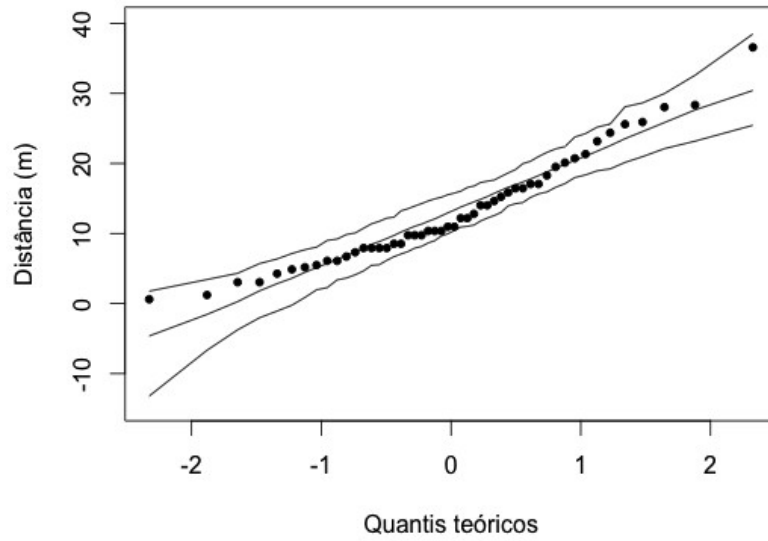
## 2. Gráfico QQ com envelope
nsim <- 100 # Número de simulações
conf <- 0.95 # Coef. de confiança

# Dados simulados ~ normal
dadossim <- matrix(rnorm(n * nsim, mean = xb, sd = sx), nrow = n)
dadossim <- apply(dadossim, 2, sort)

# Limites da banda e média
infsup <- apply(dadossim, 1, quantile, probs = c((1 - conf) / 2,
        (1 + conf) / 2))
xbsim <- rowMeans(dadossim)

# Gráfico
faixay <- range(dados, dadossim)
qq0 <- qqnorm(dados, main = "", xlab = "Quantis teóricos", pch = 20,
              ylab = "Distância (m)", ylim = faixa)
eixox <- sort(qq0$x)
lines(eixox, xbsim)
lines(eixox, infsup[1,])
lines(eixox, infsup[2,])

```



## 3. **Exercício**. Construa gráficos QQ com envelope para outras  
## distribuições.