

Análise de correspondência

(em linguagem R)

2025

Exemplo 1

Dados de oito espécies de árvores encontradas em dez diferentes locais no estado de Wisconsin, EUA estão no arquivo <https://pages.stat.wisc.edu/~rich/JWMULT06dat/T12-10.DAT> (vide também Johnson and Wichern, 2007, *Applied Multivariate Statistical Analysis*, 6th ed., p. 736).

```
## Leitura dos dados
dados <- read.table("https://pages.stat.wisc.edu/~rich/JWMULT06dat/T12-10.DAT")
tc <- as.table(as.matrix(dados[, 2:11]))
nl <- dados$V1
nc <- 1:10
rownames(tc) <- nl
colnames(tc) <- nc
tc
```

```
##           1 2 3 4 5 6 7 8 9 10
## BurOak    9 8 3 5 6 0 5 0 0 0
## BlackOak  8 9 8 7 0 0 0 0 0 0
## WhiteOak  5 4 9 9 7 7 4 6 0 2
## RedOak    3 4 0 6 9 8 7 6 4 3
## AmericanElm 2 2 4 5 6 0 5 0 2 5
## Basswood  0 0 0 0 2 7 6 6 7 6
## Ironwood  0 0 0 0 0 0 7 4 6 5
## SugarMaple 0 0 0 0 0 5 4 8 8 9
```

O teste qui-quadrado indica rejeição da hipótese de independência.

```
chisq.test(tc)
```

```
## Warning in chisq.test(tc): Chi-squared approximation may be incorrect
```

```
##
## Pearson's Chi-squared test
##
## data: tc
## X-squared = 224,85, df = 63, p-value < 2,2e-16
```

Exercício 1. Por que a mensagem “Chi-squared approximation may be incorrect” foi mostrada?

```
## Matriz de correspondência
n <- sum(tc)
cat("\n n =", n, ", I =", nrow(tc), ", J =", ncol(tc), "\n")

##
## n = 292 , I = 8 , J = 10
```

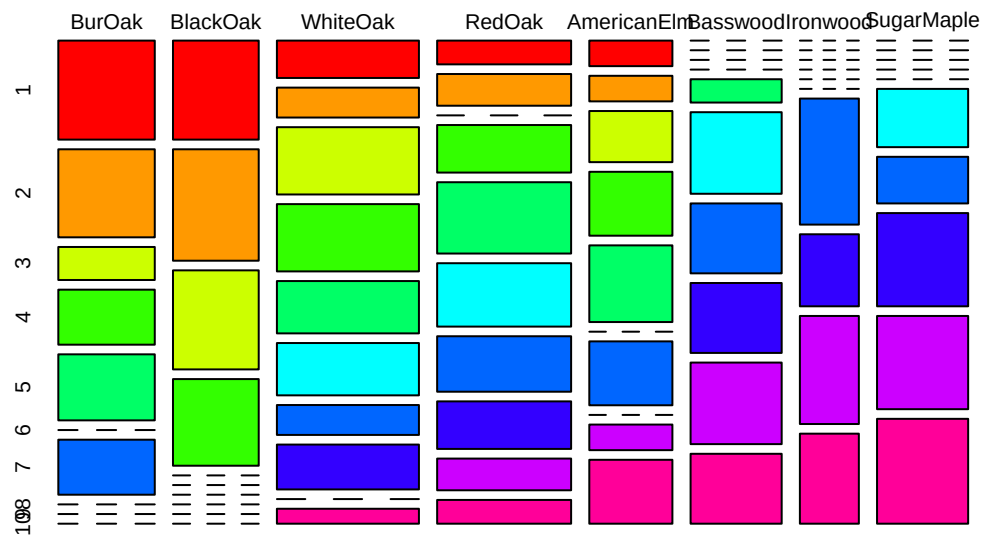
```

P <- tc / n
pr <- rowSums(P)
pc <- colSums(P)
Dr12 <- diag(1 / sqrt(pr))
Dc12 <- diag(1 / sqrt(pc))

## Gráficos de mosaico
tci <- n * as.table(crossprod(t(pr), pc)) # independência
rownames(tci) <- nl
colnames(tci) <- nc
cores <- rainbow(ncol(tc))
mosaicplot(tc, main = "Dados", col = cores)

```

Dados

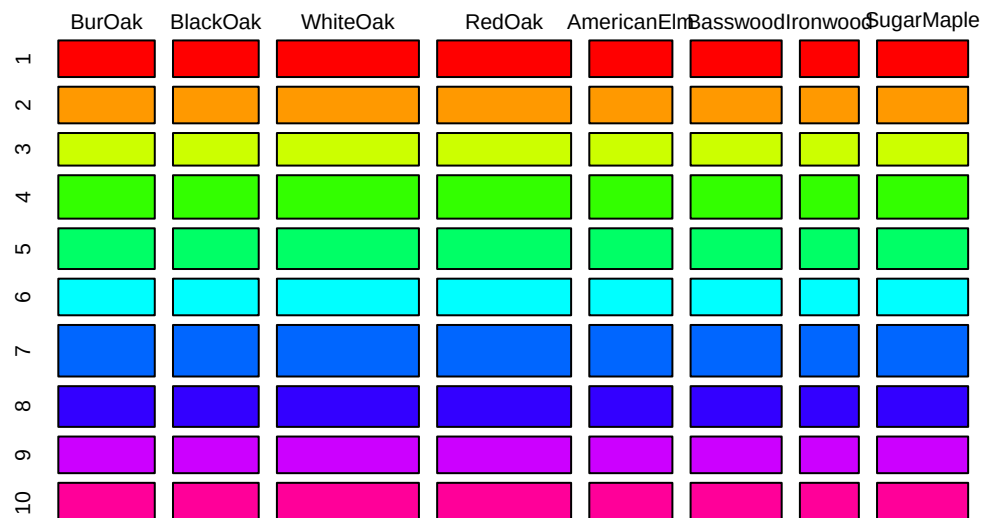


```

mosaicplot(tci, main = "Independência", col = cores)

```

Independência



```

## Matriz de resíduos e decomposição em valores singulares
Ptil <- P - pr %*% t(pc)
Ptile <- Dr12 %*% Ptil %*% Dc12
mca <- svd(Ptile)
k <- min(dim(P)) - 1
names(mca)

## [1] "d" "u" "v"

## Inércias
ine2 <- mca$d[1:2]^2 / sum(mca$d^2) * 100
cat("\n Inércias =", ine2)

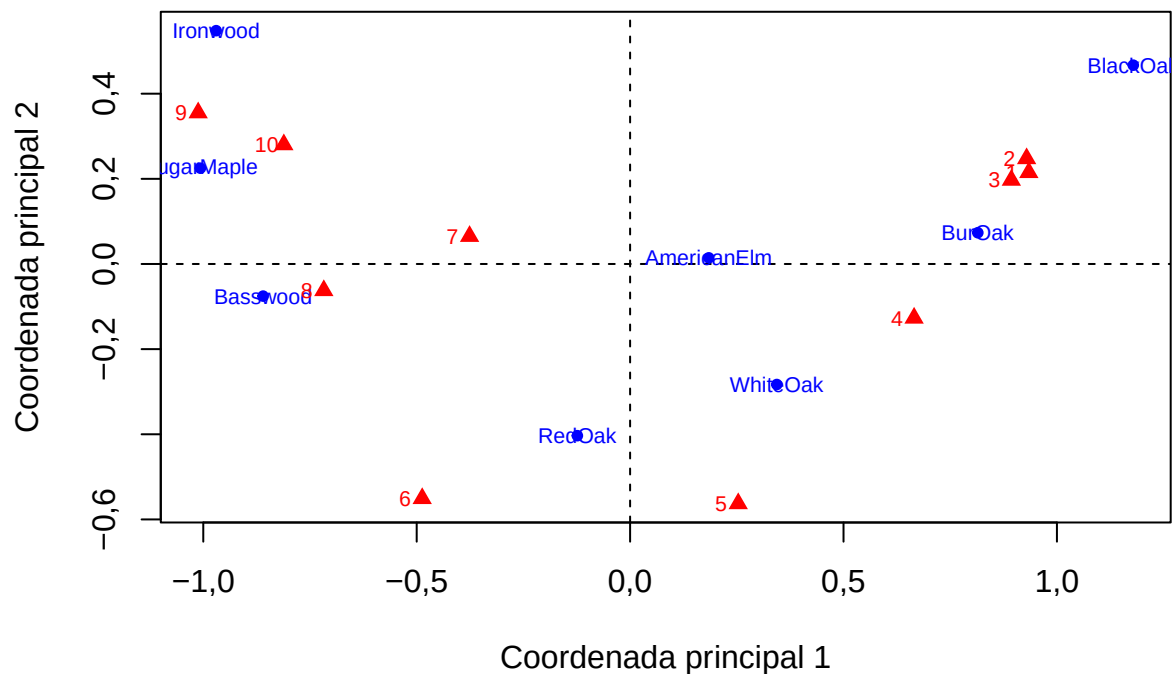
##
## Inércias = 69,7013 12,4904
cat("\n Inércias acumuladas=", cumsum(ine2))

##
## Inércias acumuladas= 69,7013 82,1917

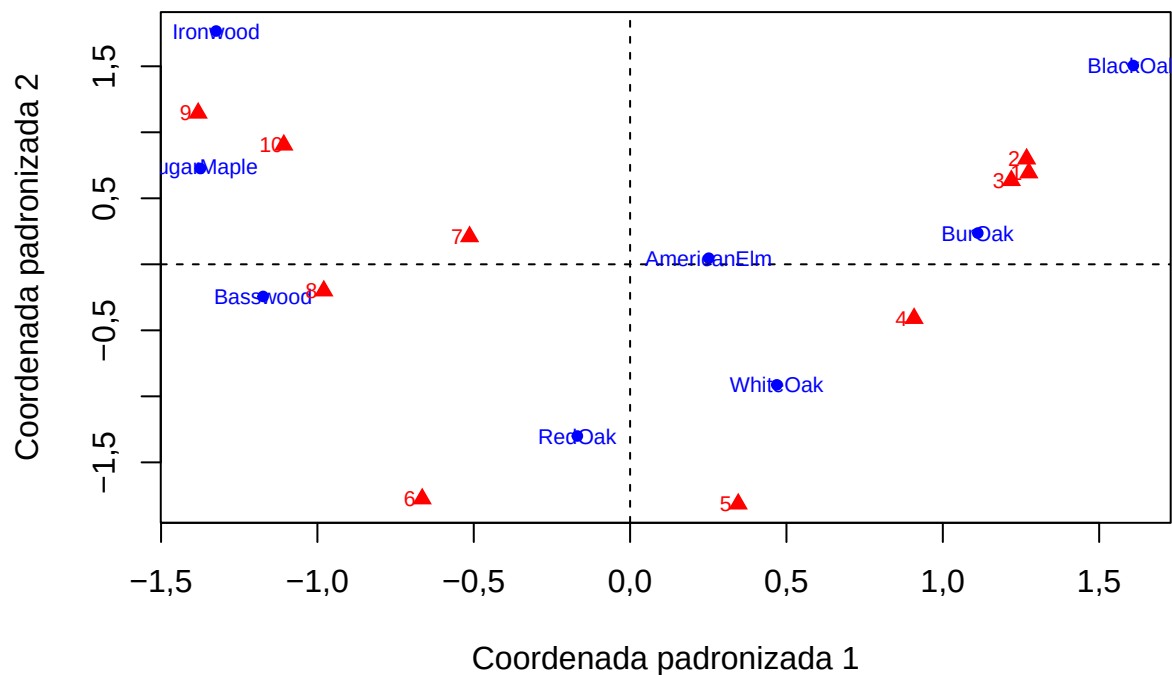
## Coordenadas padronizadas e principais
csl <- Dr12 %*% mca$u[, 1:k]
csc <- Dc12 %*% mca$v[, 1:k]
#
Lambda <- diag(mca$d[1:k])
cpl <- csl %*% Lambda
cpc <- csc %*% Lambda

## Gráficos de coordenadas principais e padronizadas
ru <- range(cpl[, 1], cpc[, 1])
rv <- range(cpl[, 2], cpc[, 2])
plot(cpl[, 1], cpl[, 2], pch = 20, xlim = ru, ylim = rv, col = "blue",
     xlab = "Coordenada principal 1", ylab = "Coordenada principal 2")
points(cpc[, 1], cpc[, 2], pch = 17, col = "red")
abline(h = 0, lty = 2)
abline(v = 0, lty = 2)
text(cpl[, 1], cpl[, 2], nl, col = "blue", cex = 0.7)
text(cpc[, 1] - 0.04, cpc[, 2], nc, col = "red", cex = 0.7)

```



```
ru <- range(csl[, 1], csc[, 1])
rv <- range(csl[, 2], csc[, 2])
plot(csl[, 1], csl[, 2], pch = 20, xlim = ru, ylim = rv, col = "blue",
      xlab = "Coordenada padronizada 1", ylab = "Coordenada padronizada 2")
points(csc[, 1], csc[, 2], pch = 17, col = "red")
abline(h = 0, lty = 2)
abline(v = 0, lty = 2)
text(csl[, 1], csl[, 2], nl, col = "blue", cex = 0.7)
text(csc[, 1] - 0.04, csc[, 2], nc, col = "red", cex = 0.7)
```



Exercício 2. Comente as associações nos dois gráficos acima.

Exercício 3. Refaça o exemplo eliminando o local identificado como “10”.

Exemplo 2

Dados de uma amostra de pessoas incluem as variáveis rendimento e satisfação com o trabalho. Os níveis de rendimento são *Low*, *Middle* e *High*. Os níveis de satisfação são *Very dissatisfied*, *Somewhat dissatisfied*, *Moderately satisfied* e *Very satisfied*, codificados como 1, 2, 3 e 4, respectivamente. Os dados estão no arquivo <https://pages.stat.wisc.edu/~rich/JWMULT06dat/T12-10.DAT> (vide também Johnson and Wichern, 2007, *Applied Multivariate Statistical Analysis*, 6th ed., p. 753).

```
## Leitura dos dados
dados <- read.table("https://pages.stat.wisc.edu/~rich/JWMULT06dat/T12-15.DAT")
tc <- as.table(as.matrix(dados[, 2:5]))
nl <- dados$V1
nc <- 1:4
rownames(tc) <- nl
colnames(tc) <- nc
tc
```

```
##          1  2  3  4
## Low      42 62 184 207
## Middle   13 28  81 113
## High      7 18  54  92
```

O teste qui-quadrado indica não rejeição da hipótese de independência.

```
chisq.test(tc)
```

```
##
## Pearson's Chi-squared test
##
## data:  tc
## X-squared = 10,407, df = 6, p-value = 0,1085
```

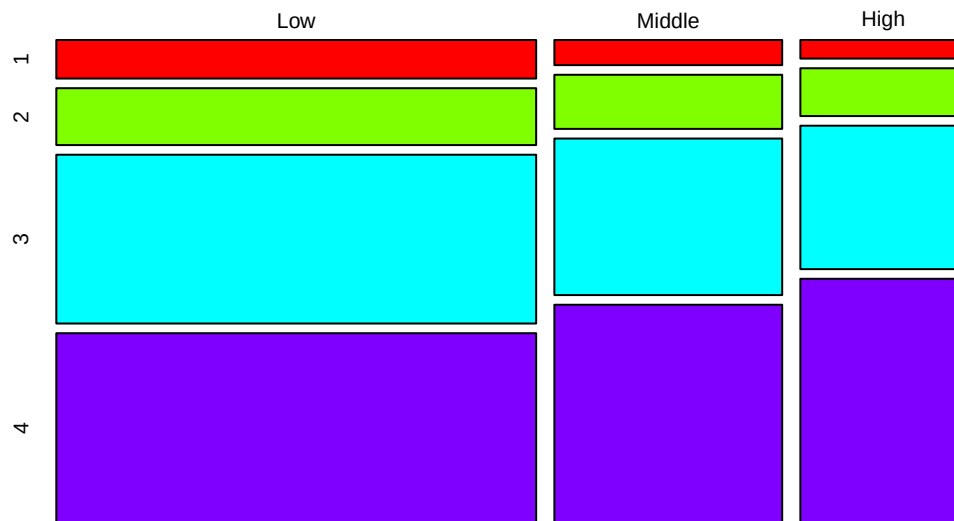
```
## Matriz de correspondência
n <- sum(tc)
cat("\n n =", n, ", I =", nrow(tc), ", J =", ncol(tc), "\n")
```

```
##
## n = 901 , I = 3 , J = 4
```

```
P <- tc / n
pr <- rowSums(P)
pc <- colSums(P)
Dr12 <- diag(1 / sqrt(pr))
Dc12 <- diag(1 / sqrt(pc))
```

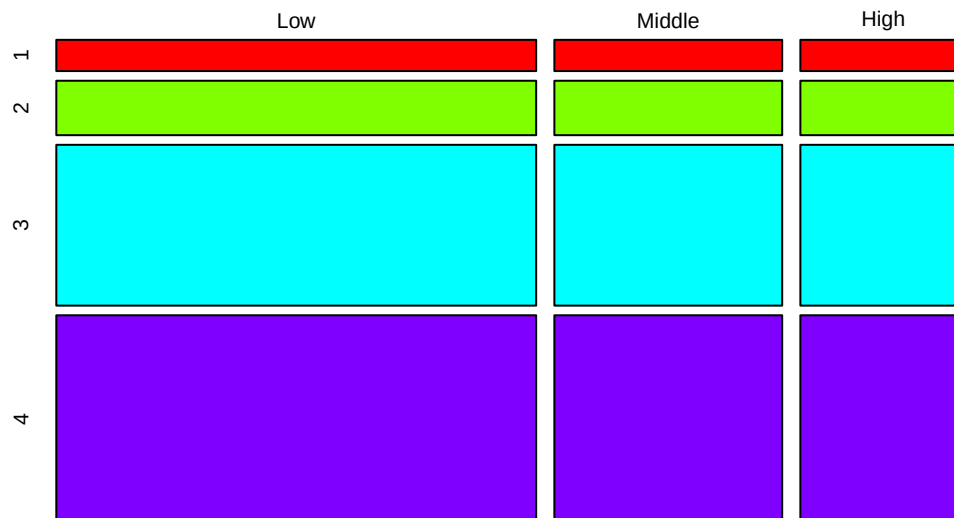
```
## Gráficos de mosaico
tci <- n * as.table(crossprod(t(pr), pc)) # independência
rownames(tci) <- nl
colnames(tci) <- nc
cores <- rainbow(ncol(tc))
mosaicplot(tc, main = "Dados", col = cores)
```

Dados



```
mosaicplot(tci, main = "Independência", col = cores)
```

Independência



```
## Matriz de resíduos e decomposição em valores singulares
Ptil <- P - pr %*% t(pc)
Ptile <- Dr12 %*% Ptil %*% Dc12
mca <- svd(Ptile)
k <- min(dim(P)) - 1
```

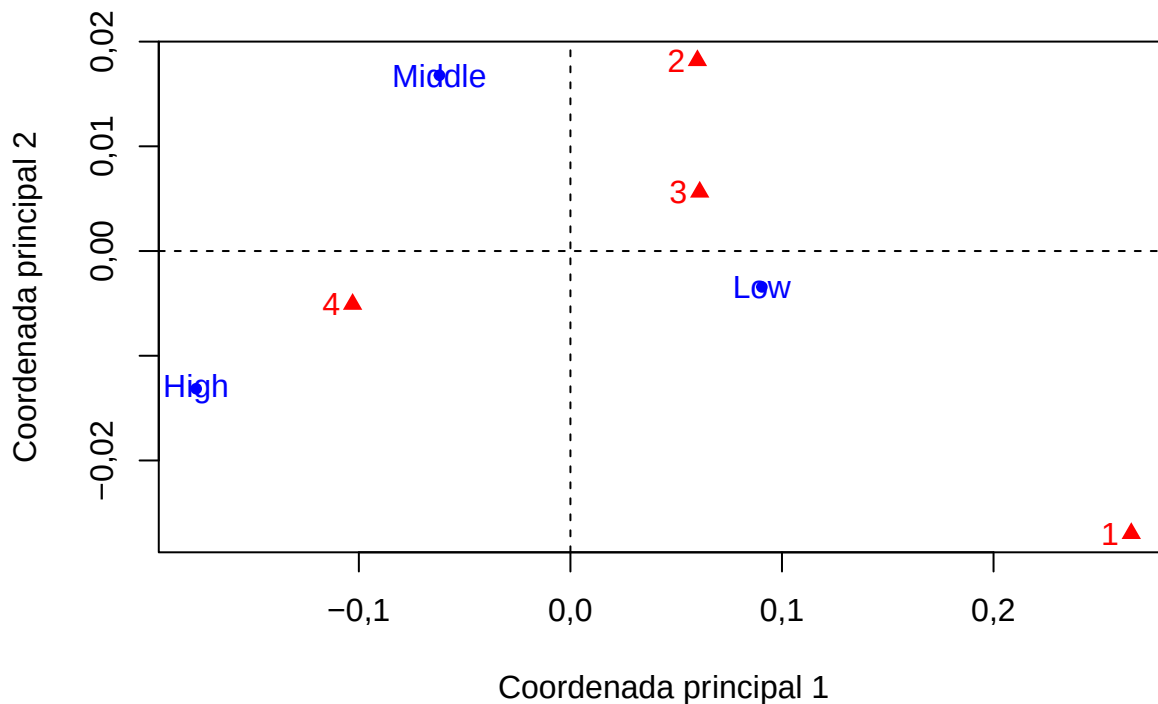
```
## Inércias
ine2 <- mca$d[1:2]^2 / sum(mca$d^2) * 100
cat("\n Inércias =", ine2)
```

```
##
## Inércias = 99,02381 0,9761874
```

```
cat("\n Inércias acumuladas=", cumsum(ine2))

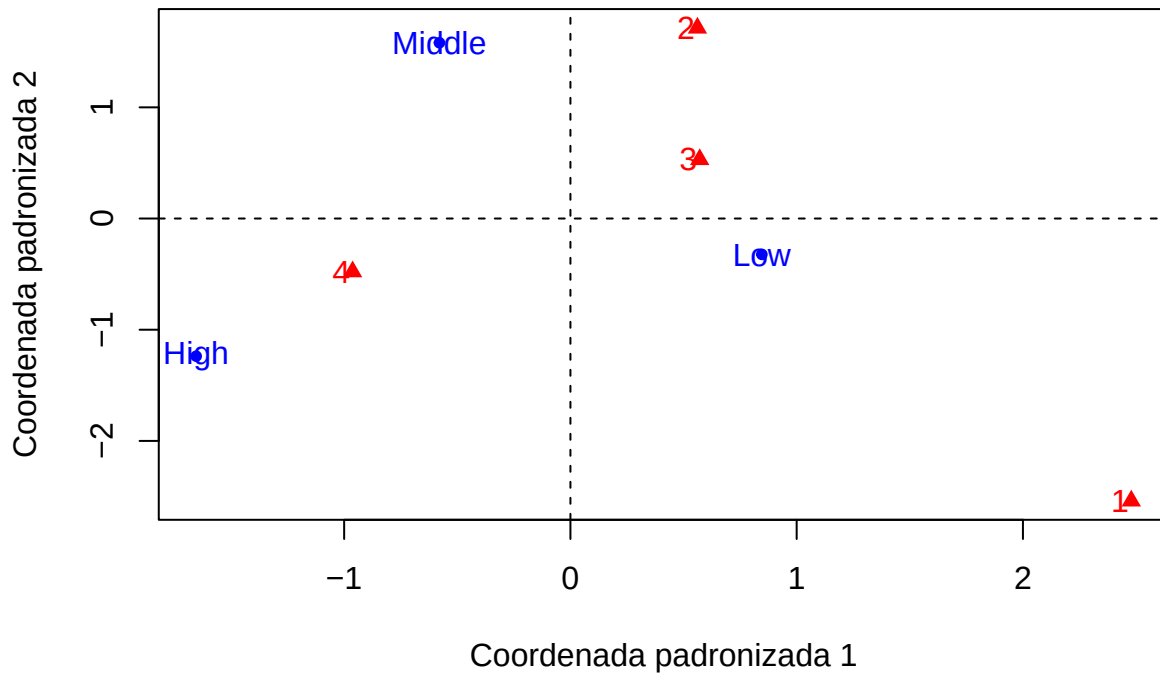
##
## Inércias acumuladas= 99,02381 100
## Coordenadas padronizadas e principais
csl <- Dr12 %%% mca$u[, 1:k]
csc <- Dc12 %%% mca$v[, 1:k]
#
Lambda <- diag(mca$d[1:k])
cpl <- csl %%% Lambda
cpc <- csc %%% Lambda

## Gráficos de coordenadas principais e padronizadas
ru <- range(cpl[, 1], cpc[, 1])
rv <- range(cpl[, 2], cpc[, 2])
plot(cpl[, 1], cpl[, 2], pch = 20, xlim = ru, ylim = rv, col = "blue",
      xlab = "Coordenada principal 1", ylab = "Coordenada principal 2")
points(cpc[, 1], cpc[, 2], pch = 17, col = "red")
abline(h = 0, lty = 2)
abline(v = 0, lty = 2)
text(cpl[, 1], cpl[, 2], nl, col = "blue")
text(cpc[, 1] - 0.01, cpc[, 2], nc, col = "red")
```

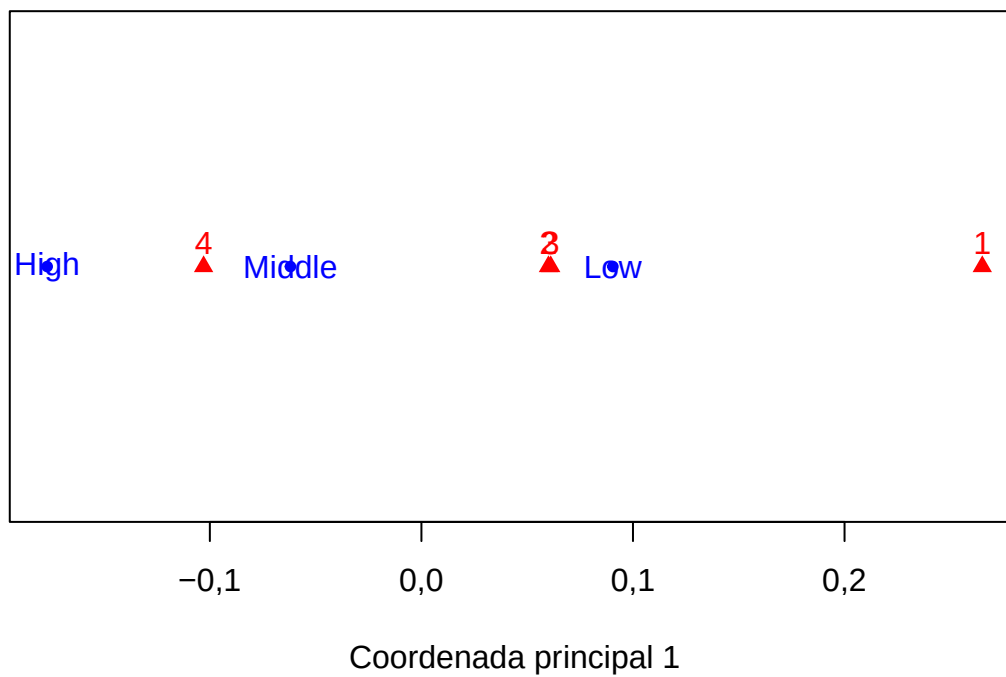


```
ru <- range(csl[, 1], csc[, 1])
rv <- range(csl[, 2], csc[, 2])
plot(csl[, 1], csl[, 2], pch = 20, xlim = ru, ylim = rv, col = "blue",
      xlab = "Coordenada padronizada 1", ylab = "Coordenada padronizada 2")
points(csc[, 1], csc[, 2], pch = 17, col = "red")
abline(h = 0, lty = 2)
abline(v = 0, lty = 2)
text(csl[, 1], csl[, 2], nl, col = "blue")
```

```
text(csc[, 1] - 0.05, csc[, 2], nc, col = "red")
```



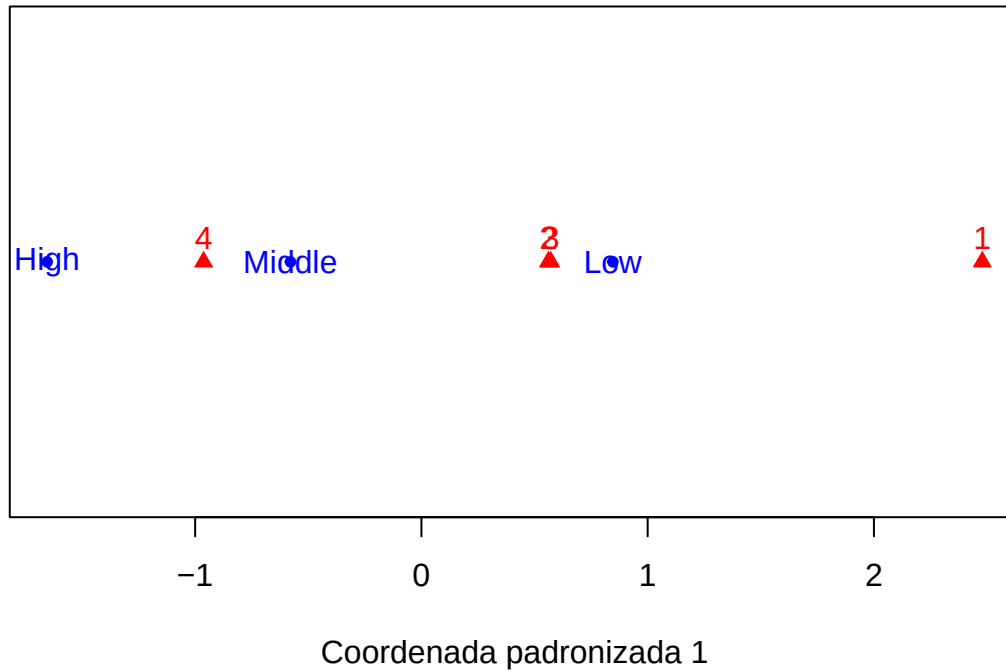
```
## Gráficos de coordenadas principais e padronizadas
ru <- range(cpl[, 1], cpc[, 1])
stripchart(cpl[, 1], method = "stack", pch = 20, col = "blue", xlim = ru,
           xlab = "Coordenada principal 1")
stripchart(cpc[, 1], method = "stack", pch = 17, col = "red", add = TRUE)
text(cpl[, 1], 1, nl, col = "blue")
text(cpc[, 1], 1.1, nc, col = "red")
```



```

ru <- range(csl[, 1], csc[, 1])
stripchart(csl[, 1], method = "stack", pch = 20, col = "blue", xlim = ru,
           xlab = "Coordenada padronizada 1")
stripchart(csc[, 1], method = "stack", pch = 17, col = "red", add = TRUE)
text(csl[, 1], 1, nl, col = "blue")
text(csc[, 1], 1.1, nc, col = "red")

```



Em linguagem R, as funções `ca` (pacote `ca`) e `corresp` (pacote `MASS`), por exemplo, implementam análise de correspondência.

Exercício 4. Refaça os exemplos em linguagem Python.