Instituto de Matemática e Estatística
Universidade de São Paulo

Lista 5

MAE0330 - Análise Multivariada de Dados

Profª Lucia Pereira Barroso

Bruno Groper Morbin - nºUSP 11809875
Luigi Pavarini de Lima - nºUSP 11844642

São Paulo
24 outubro, 2022

1 Escalonamento Multidimensional

Conjunto de dados:

```
# A tibble: 19 x 8
   Nome        Modelo                                cilindrada desempenho consumo autonomia potencia aceleracao
   <chr>       <chr>                                      <dbl>      <dbl>   <dbl>     <dbl>    <dbl>      <dbl>
 1 Onix        Chevrolet Onix Sedan LT 1.0 Turbo            999        187    13.7       603      116        9.7
 2 Sandero     Renault Sandero RS 2.0                      1998        202     9.9       495      145        8
 3 Peugeot     Peugeot 208 GT 1.6 Turbo                    1598        222    12         660      166        7.6
 4 Up          Volkswagen Up Connect 1.0 TSi                999        183    14.1       705      101        9.4
 5 HB20        Hyundai HB20 Vision 1.6                     1591        190    12.5       625      123        9.3
 6 Argo        Fiat Argo Drive S-Design 1.3                1332        184    12.5       600      101       10.8
 7 Chery       Chery Arrizo 5 RT 1.5 Turbo                 1499        190    11         528      147        9.9
 8 JAC         JAC T40 1.5                                 1499        191    11.9       500      125        9.8
 9 March       Nissan March SV 1.6                         1598        182    12.6       517      111        9.3
10 Ka          Ford Ka SE Plus 1.5                         1497        181    12.4       632      128        9.9
11 Corolla     Toyota Corolla XEi 2.0                      1987        208    11.6       580      169        9.2
12 Civic       Honda Civic EXL 2.0 AT                      1997        195    10.5       588      150       10.9
13 Azera       Hyundai Azera 3.0 V6                        2999        235     8.2       574      261        7.9
14 Fusion      Fusion SEL 2.0 Turbo FWD                    1999        195     8.6       533      248        7.2
15 Ferrari     Ferrari 812 Superfast                       6496        340     4.9       451      800        2.9
16 Porsche     Porsche Carrera GT 5.7 V10                  5733        330     2.7       248      612        3.5
17 Jaguar      Jaguar F-Pace SVR 5.0 V8                    5000        283     6.2       508      550        4.3
18 Rolls       Rolls-Royce Wraith 6.6 V12                  6592        250     4.7       390      632        4.6
19 Lamborghini Lamborghini Aventador S LP 740-4 6.5 V12    6498        350     4.5       383      740        2.9
```

Para o escalonamento multidimensional, precisamos primeiro obter a matriz de distâncias.

1.1 Matriz de distâncias

Conforme vimos na Lista 1, as variáveis deste banco de dados possuem magnitudes diferentes e por isso trabalharemos com as variáveis padronizadas. A seguir, calculamos a matriz de distância euclidiana entre as observações para o conjunto de dados contemplando apenas as variáveis qualitativas, denominado `carros1`.

```
> mdist <- dist(scale(carros1),method = "euclidean")
```

```
          1         2         3         4         5         6         7         8         9        10        11        12        13        14        15        16        17        18
2  1.6898329
3  1.2662394 1.6820612
4  0.9520413 2.4019663 1.2592207
5  0.5151364 1.5111246 0.9259826 0.9227618
6  0.5597184 1.6631299 1.4945502 1.1927477 0.6253562
7  1.0700352 0.8827893 1.6054454 1.8837574 1.0155803 0.8829158
8  1.1039429 0.9302863 1.7679368 2.0065525 1.1728970 1.0169037 0.3756611
9  0.9126604 1.0187818 1.6354919 1.7985400 1.0005552 0.9496328 0.5564227 0.3577855
10 0.5337489 1.6630681 1.1547332 0.8870149 0.2842125 0.4664246 1.0471488 1.2303111 1.0810268
11 0.9213864 1.0269771 0.9955098 1.5240162 0.6395013 0.8949544 0.6970253 0.8777429 0.8455449 0.8081925
12 1.1474406 1.3799655 1.5332110 1.6794782 0.9179761 0.7208303 0.7236684 1.0225201 1.1135398 0.8465465 0.7422057
13 2.2281603 1.2482142 1.5718869 2.6016726 1.8414684 2.1392055 1.6508131 1.8691131 1.9518135 1.9939657 1.3324765 1.6109635
14 1.9773430 0.7343941 1.6384035 2.4956647 1.6873364 1.9657380 1.3005497 1.4848971 1.5227262 1.8268147 1.2689499 1.5985482 0.9745124
15 6.0595932 4.8589944 5.2442827 6.4005985 5.7545932 6.0849318 5.4927408 5.5718055 5.6640185 5.9365996 5.2249365 5.5818785 4.0906313 4.5358530
16 6.4259205 4.9059475 5.8739193 7.0060856 6.1930301 6.3952727 5.6209243 5.6318794 5.7926093 6.3663176 5.5901169 5.8749845 4.5385970 4.7549002 2.1599641
17 4.3654623 3.1465526 3.5379812 4.7003502 4.0288221 4.3857597 3.8001299 3.9089707 3.9790452 4.2129134 3.5175260 3.8904251 2.3476020 2.7764619 1.8057137 2.7595195
18 5.1986005 3.7966103 4.6352948 5.6712215 4.8959208 5.1471538 4.4553248 4.5213550 4.6178601 5.0426593 4.3527768 4.6093361 3.2818652 3.5063746 1.9143400 2.0778480 1.5526747
19 6.2498515 4.9437380 5.5019946 6.6702957 5.9687521 6.2645020 5.6148416 5.6647255 5.7784781 6.1556962 5.4057766 5.7567970 4.2882199 4.6911509 0.7017960 1.5423099 2.0803199 1.9121826
```

1.2 Método

Avaliamos a qualidade da representação em K dimensões através da métrica GOF (bondade do ajuste):

$$GOF_K = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{n} |\lambda_i|},$$

por valor absoluto onde K refere-se ao número da dimensão escolhida, e n é o número de observações (total de dimensão).

$$GOF_K = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{n} max(\lambda_i, 0)}$$

por valor no máximo.

```
        Absoluto    Máximo

k=1  0.9046937   0.9046937

k=2  0.9647328   0.9647328

k=3  0.9799292   0.9799292
```

```
k=4  0.9901546   0.9901546
```

Verificamos então que para apenas 1 dimensão já temos mais de 90% de bondade de ajuste (variáveis explicadas pela configuração). Escolhemos então a configuração com 2 dimensões para fins de melhor visualização, além de apresentar em torno de 96% de bondade de ajuste.

Nota: GOF por valor absoluto e no máximo coincidiram devido à precisão dos autovalores. Segue exemplo para K=2:

```
> EMC <- cmdscale(mdist, eig=TRUE, k=2) # k é o número da dimensão
> round(EMC$eig,4) # mostra os autovalores
```

```
 [1] 97.7069  6.4842  1.6412  1.1043  0.9131  0.1502  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
[13]  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
```

```
> EMC$GOF # mostra bondade do ajuste, com base no valor absoluto e no máximo
```

```
        Absoluto    Máximo
k=1  0.9046937   0.9046937
```

1.3 EM 2D

Coordenadas obtidas pelo método de escalonamento multidimensional com 2 dimensões:

```
           V1          V2
1  -1.95726036 -0.07444340
2  -0.52483625  0.62751300
3  -1.16753984 -0.91095392
4  -2.35533760 -0.89053070
5  -1.67308705 -0.28521294
6  -1.95130871  0.12974848
7  -1.23889462  0.59163868
8  -1.26756277  0.77535149
9  -1.40004472  0.61071120
10 -1.85088265 -0.22996310
11 -1.11063541 -0.05768787
12 -1.38926444  0.18810193
13  0.07008482 -0.26285116
14 -0.24802231  0.24017681
15  4.03586036 -0.92143135
16  4.33115390  1.13174571
17  2.32976240 -0.66258508
18  3.08878764  0.35558864
19  4.27902762 -0.35491642
```

1.3.1 Visualização

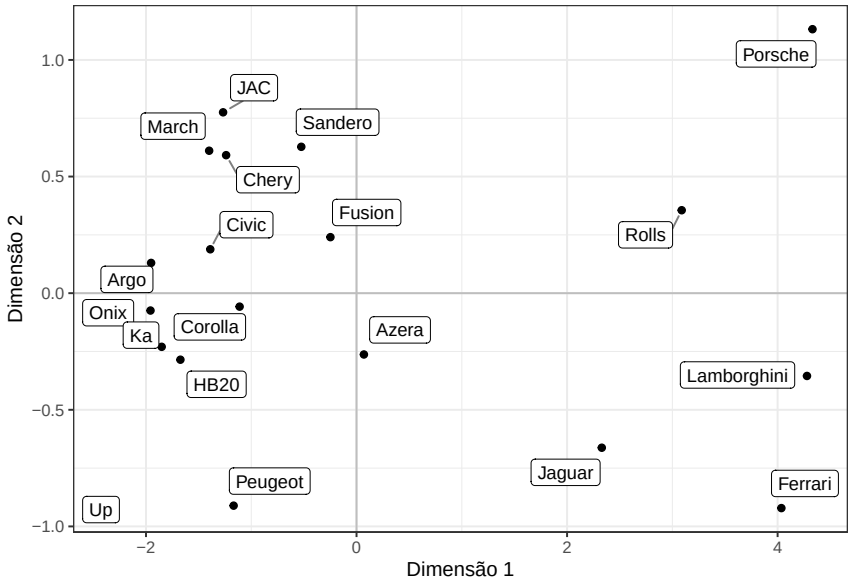


Figura 1: Coordenadas por escalonamento multidimensional para 2 dimensões.

1.3.2 Agrupamentos

Conforme visualizamos abaixo, temos a seguinte separação de grupos dos carros a partir do algoritmo de k-médias de acordo com as coordenadas obtidas a partir do escalonamento multidimensional. Já no segundo gráfico abaixo, assim como obtivemos na Lista 2, podemos observar o agrupamento formado pelo método de vizinhos mais próximos (*"single-linkage"*).

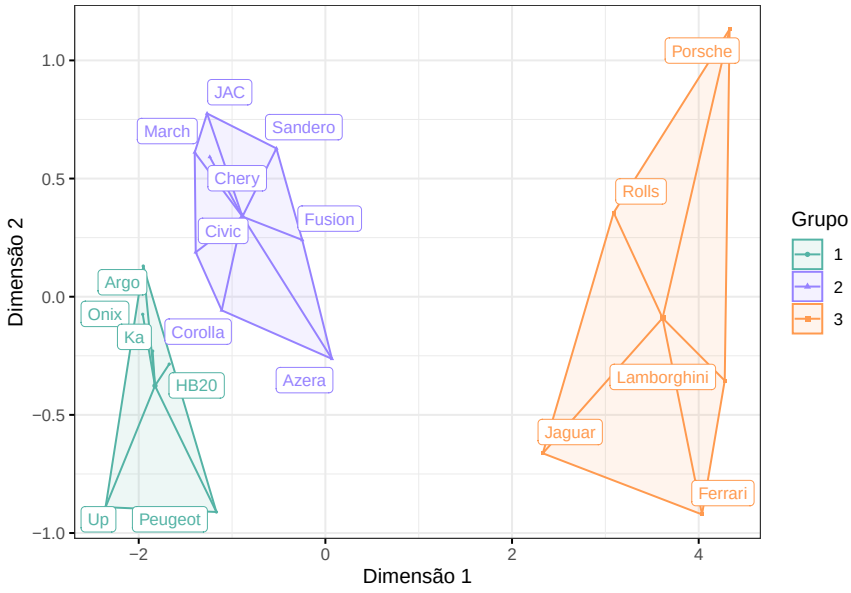


Figura 2: Gráfico K-médias.

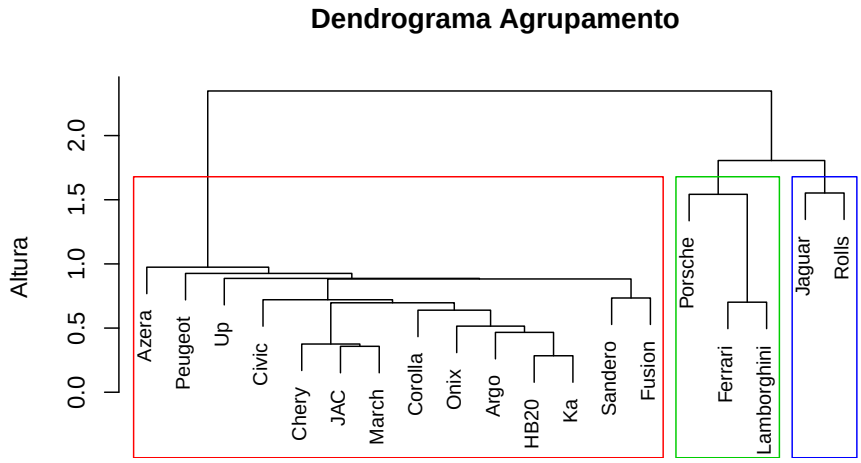


Figura 3: Gráfico Vizinho mais próximo.

Observamos então que a principal diferença entre o agrupamento obtido na lista 2 com o método de k-médias foi que na lista 2 houve uma divisão entre carros populares e carros esportivos de forma que dentro de carros esportivos foi-se formado dois grupos. Já no método de k-médias pelas coordenadas observadas, houve a divisão de um grande grupo de carros esportivos (como se tivessemos juntado os dois grupos esportivos do agrupamento anterior), e dois grupos de carros populares que pode ser considerado uma fragmentação do terceiro grupo do agrupamento anterior.

2 PCA (Análise de Componentes Principais)

Como alternativa para o escalonamento multidimensional, toma-se uma segunda configuração baseada no método de Componentes Principais, obtendo os escores das observações.

```
> CP <- princomp(scale(carros1)) # Aplicando nos dados quantitativos padronizados
> as.data.frame(CP$scores[,1:2])
```

```
        Comp.1      Comp.2
1  -1.95726036  0.07444340
2  -0.52483625 -0.62751300
3  -1.16753984  0.91095392
4  -2.35533760  0.89053070
5  -1.67308705  0.28521294
6  -1.95130871 -0.12974848
7  -1.23889462 -0.59163868
8  -1.26756277 -0.77535149
9  -1.40004472 -0.61071120
10 -1.85088265  0.22996310
11 -1.11063541  0.05768787
12 -1.38926444 -0.18810193
13  0.07008482  0.26285116
14 -0.24802231 -0.24017681
15  4.03586036  0.92143135
16  4.33115390 -1.13174571
17  2.32976240  0.66258508
18  3.08878764 -0.35558864
19  4.27902762  0.35491642
```

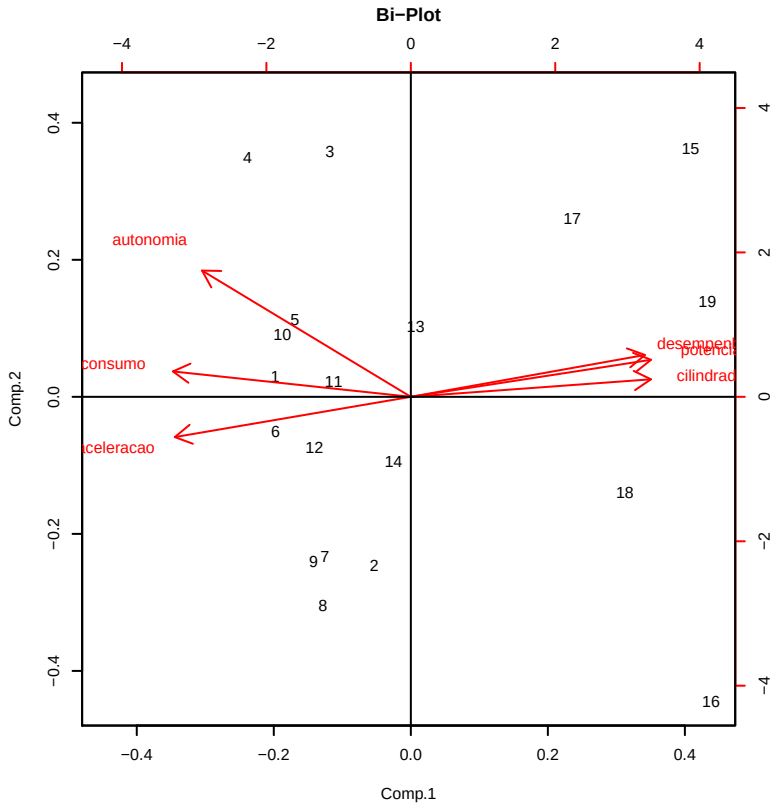
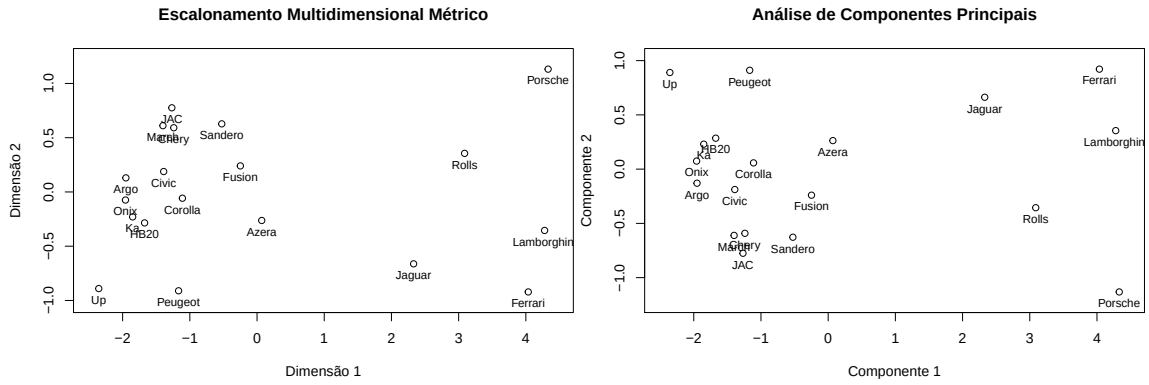


Figura 4: Bi-Plot com PCA.

3 Procrustes



(a) Coordenadas por EM.

(b) Coordenadas por PCA.

Figura 5: Comparação das coordenadas pelas duas configurações.

```
> ## ANÁLISE PROCRUSTES
> library(vegan)
>
> X <- EMC$points
> Y <- CP$scores[,1:2]
>
> proc <- procrustes(X, Y)
>
> summary(proc)
```

```
Call:
procrustes(X = X, Y = Y)

Number of objects: 19    Number of dimensions: 2

Procrustes sum of squares:
        0
Procrustes root mean squared error:
        0
Quantiles of Procrustes errors:
         Min           1Q       Median           3Q          Max
2.636780e-16 8.301368e-16 1.421779e-15 1.915135e-15 3.016049e-15

Rotation matrix:
             [,1]          [,2]
[1,] 1.000000e+00  2.150014e-17
[2,] 2.150014e-17 -1.000000e+00

Translation of averages:
              [,1]          [,2]
[1,] -4.099654e-16 -7.647277e-16

Scaling of target:
[1] 1
```

```
> # teste de não aleatoriedade das duas configurações
> protest(X, Y)
```

```
Call:
protest(X = X, Y = Y)

Procrustes Sum of Squares (m12 squared):           0
Correlation in a symmetric Procrustes rotation:    1
Significance:  0.001

Permutation: free
Number of permutations: 999
```

Obtemos que a soma de quadrados de procrustes é igual a zero, indicando um ajuste praticamente perfeito, uma vez que podemos notar pelos gráficos de PCA e EM Métrico que, de fato, estamos avaliando o mesmo gráfico mas “espelhado” pelo horizonte. Nesse mesmo sentido, a indicação do output do comando “protest(.)” é de rejeição da hipótese nula, que avalia a aleatoriedade da diferença das coordenadas. Ou seja, de forma breve, podemos afirmar que ao nível de significância de 0.1% podemos considerar que ambas análises são iguais.

4 Apêndice

4.1 Código

```
> carros <- read_excel("carros.xls")
> carros1 <- carros[,3:ncol(carros)]
> carros
>
> mdist <- dist(scale(carros1),method = "euclidean")
>
> cat("    \tAbsoluto","\tMáximo\n")
> cat("k=1\t",cmdscale(mdist, eig=TRUE, k=1)$GOF[[1]],"\t",cmdscale(mdist, eig=TRUE, k=1)$GOF[[2]],"\n")
> cat("k=2\t",cmdscale(mdist, eig=TRUE, k=2)$GOF[[1]],"\t",cmdscale(mdist, eig=TRUE, k=2)$GOF[[2]],"\n")
> cat("k=3\t",cmdscale(mdist, eig=TRUE, k=3)$GOF[[1]],"\t",cmdscale(mdist, eig=TRUE, k=3)$GOF[[2]],"\n")
> cat("k=4\t",cmdscale(mdist, eig=TRUE, k=4)$GOF[[1]],"\t",cmdscale(mdist, eig=TRUE, k=4)$GOF[[2]],"\n")
>
> EMC <- cmdscale(mdist, eig=TRUE, k=2) # k é o número da dimensão
> round(EMC$eig,4) # mostra os autovalores
> EMC$GOF # mostra bondade do ajuste, com base no valor absoluto e no máximo
> cat("    \tAbsoluto","\tMáximo\n")
> cat("k=1\t",cmdscale(mdist, eig=TRUE, k=1)$GOF[[1]],"\t",cmdscale(mdist, eig=TRUE, k=1)$GOF[[2]],"\n")
>
> as.data.frame(EMC$points) # mostra as coordenadas
>
> # Plot com ggplot
> library(ggplot2)
> library(ggrepel)
> coord1 <- data.frame(EMC$points)
>
> ggplot(coord1, aes(x=X1, y=X2))+
>   geom_hline(yintercept = 0,col="grey")+
>   geom_vline(xintercept = 0,col="grey")+
>   geom_point() +
>   geom_label_repel(aes(label=carros$Nome), size=3.5, box.padding = 0.35,
>                    point.padding = 0.5, segment.color = 'grey50') +
>   theme_bw()+
>   labs (x= "Dimensão 1", y = "Dimensão 2")
>
> # agrupamentos
> library(dplyr)
> library(ggpubr)
> EMA <- as.data.frame(carros1) %>% scale() %>% dist() %>% cmdscale() %>% as_tibble()
> colnames(coord1) <- c("Dimensão 1", "Dimensão 2")
> coord1$Grupo <- kmeans(coord1, centers = 3, nstart=100)$cluster %>% as.factor()
>
> rownames(coord1) <- carros$Nome
> ggscatter(coord1, x="Dimensão 1", y = "Dimensão 2",
>           label = rownames(coord1),
>           font.label = c(9, "plain"),
>           label.rectangle = TRUE,
>           show.legend.text = FALSE,
>           color = "Grupo",
>           palette = c("#54B3A5", "#9682FF", "#FF9A4F"),
>           size=0.5,
>           shape = "Grupo",
>           ellipse = "TRUE",
>           ellipse.type = "convex",
>           mean.point = TRUE,
>           star.plot = TRUE,
```

```
>           repel = TRUE,
>           ggtheme = theme_bw())
>
> library(cluster)
> hc3<-hclust(mdist,method = "single")
> plot(hc3, labels = carros$Nome, main = "Dendrograma Agrupamento", xlab=NA, ylab="Altura", sub=NA, cex=0.8)
> rect.hclust(hc3 , k = 3, border = 2:6)
>
> CP <- princomp(scale(carros1)) # Aplicando nos dados quantitativos padronizados
> as.data.frame(CP$scores[,1:2])
>
> par(cex=0.5,xaxs="r")
> par(mar = c(5, 5, 5, 5))
> biplot(CP,cex = 1, main="Bi-Plot")
> abline(a=0,b=0)
> abline(v=0)
>
> # Plot por EM 2D
> par(cex=0.9)
> x <- EMC$points[,1]
> y <- EMC$points[,2]
> plot(x, y,
>      xlim=c(range(x)[[1]]-0.1,range(x)[[2]]+0.1),
>      ylim=c(range(y)[[1]]-0.1,range(y)[[2]]+0.1),
>      xlab="Dimensão 1", ylab="Dimensão 2", main="Escalonamento Multidimensional Métrico",
>      type="point")
> text(x, y, labels = carros$Nome, cex=.8,pos = 1)
>
> # Plot por PCA
> par(cex=0.9)
> x <- CP$scores[,1]
> y <- CP$scores[,2]
> plot(x, y,
>      xlim=c(range(x)[[1]]-0.1,range(x)[[2]]+0.1),
>      ylim=c(range(y)[[1]]-0.1,range(y)[[2]]+0.1),
>      xlab="Componente 1", ylab="Componente 2", main="Análise de Componentes Principais",
>      type="point")
> text(x, y, labels = carros$Nome, cex=.8,pos = 1)
>
> ## ANÁLISE PROCRUSTES
> library(vegan)
>
> X <- EMC$points
> Y <- CP$scores[,1:2]
>
> proc <- procrustes(X, Y)
>
> summary(proc)
>
> proc$Yrot
> proc$X
>
> # distância procrustes
> proc$ss
>
> plot(proc)
> plot(proc, kind=2)
> residuals(proc)
>
```

```
> # teste de não aleatoriedade das duas configurações
> protest(X, Y)
```