Instituto de Matemática e Estatística
Universidade de São Paulo

Lista 7

MAE0330 - Análise Multivariada de Dados

Profª Lucia Pereira Barroso

Bruno Groper Morbin - nºUSP 11809875

Luigi Pavarini de Lima - nºUSP 11844642

São Paulo
18 novembro, 2022

Análise Descritiva

Primeiramente, realiza-se uma análise descritiva sobre as variáveis quantitativas do conjunto de dados para avaliar se é coerente ou não utilizar os dados padronizados. O conjunto de dados utilizado pode ser obtido através do arquivo `trigo.xls`.

```
> # Lendo o arquivo
> trigo <- read_excel("trigo.xls")
>
> # Atribuindo estrutura fatorial para as colunas das variáveis respostas: localidade e variedade
> trigo$Localidade <- as.factor(trigo$Localidade)
> trigo$Variedade <- as.factor(trigo$Variedade)
>
> # Transformando em dataframe para realizar a análise
> trigo <- as.data.frame(trigo)
>
> # Criando a coluna de população referente à junção das variáveis respostas
> trigo1 <- trigo%>%mutate(Populacao=case_when(
+   (Localidade == 1 & Variedade == "Arkan") ~ "P1",
+   (Localidade == 1 & Variedade == "Arthur") ~ "P2",
+   (Localidade == 2 & Variedade == "Arkan") ~ "P3",
+   (Localidade == 2 & Variedade == "Arthur") ~ "P4"
+ ))%>%dplyr::select(-Localidade,-Variedade)%>%dplyr::select(Populacao,everything())
>
> # mostrando estrutura dos dados no R
> str(trigo)
```

```
'data.frame':   172 obs. of  10 variables:
 $ Localidade   : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Variedade    : Factor w/ 2 levels "Arkan","Arthur": 1 1 1 1 1 1 1 1 1 1 ...
 $ Area_b       : num  2965 3041 2907 2728 2658 ...
 $ Perimetro_b  : num  219 221 223 212 207 203 208 218 216 219 ...
 $ Comprimento_b: num  89 91 90 87 78 82 84 91 81 93 ...
 $ Largura_b    : num  43 46 44 41 42 41 40 41 44 40 ...
 $ Area_a       : num  3204 3165 3035 2867 2807 ...
 $ Perimetro_a  : num  226 224 223 215 211 207 211 216 221 226 ...
 $ Comprimento_a: num  89 91 91 88 81 82 84 90 86 93 ...
 $ Largura_a    : num  47 46 44 44 44 42 42 41 50 46 ...
```

Pode-se observar portanto que há uma diferença na ordem de grandeza das variáveis como a variável `Area_b` com números em milhares enquanto que `Largura_b` oscila em números decimais. Por esse motivo, decide-se então inicialmente padronizar os dados para realizar a análise, fazendo com que o ajuste do modelo não tenha forte influência da ordem de medida das variáveis preditoras.

Para futuramente realizar as previsões, guardam-se os valores das médias amostrais e desvios padrão amostrais para padronizar os futuros novos dados amostrais.

Médias amostrais:

Area_b	Perimetro_b	Comprimento_b	Largura_b	Area_a	Perimetro_a	Comprimento_a	Largura_a
2960.19186	220.06977	90.08140	44.83721	2906.35465	220.17442	90.26744	43.53488

Desvios padrão amostrais:

Area_b	Perimetro_b	Comprimento_b	Largura_b	Area_a	Perimetro_a	Comprimento_a	Largura_a
243.590451	8.491882	5.518104	6.422476	240.827052	9.309734	4.708893	2.781298

Análise Discriminante

Prossegue-se com a análise discriminante considerando o conjunto de dados padronizados. Para ajuste do modelo, verificará-se a seguir quais das duas funções (linear ou quadrática) performam melhor.

Toma-se os seguintes rótulos para a variável resposta:

- **P1** ≡ Localidade 1 e Variedade Arkan
- **P2** ≡ Localidade 1 e Variedade Arthur
- **P3** ≡ Localidade 2 e Variedade Arkan
- **P4** ≡ Localidade 2 e Variedade Arthur

Função linear

Nota-se que o conjunto amostral apresenta um desbalanceamento das populações, apresentando uma priori amostral não balanceada.

Tabela de contigência das variáveis Localidade e Variedade da amostra para ajuste:

```
          Variedade
Localidade Arkan Arthur
         1    36     36
         2    50     50
```

Priori amostral:

```
  P1   P2   P3   P4
0.21 0.21 0.29 0.29
```

Visto que o ajuste no programa R com a função `lda(.)` usa priori amostral, decidi-se inserir priori não informativa para realização da análise discriminatória passando um vetor balanceado representando priori de 25% para cada população no argumento `prior` da função.

Tem-se o seguinte ajuste utilizando função linear para a análise discriminante:

```
> adl <- lda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1pad[,1], CV=F, prior=rep(1/4,4))
```

Para verificar o desempenho do ajuste no próprio conjunto amostral de treino, usa-se a função `lda(.)` com o argumento de *cross-validation* verdadeiro: `CV=TRUE`. Com isso, pode-se observar através da matriz de confusão os acertos e erros do ajuste sobre a própria amostra utilizada para o mesmo.

```
> adl1 <- lda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))
>
> #matriz de confusão
> cvl1<-table(trigo1$Populacao,adl1$class)
```

Tabela 1: Matriz de confusão para o ajuste linear.

	P1	**P2**	**P3**	**P4**
P1	24	1	11	0
P2	6	15	2	13
P3	12	1	35	2
P4	1	12	6	31

```
Proporção de acerto para cada grupo:
```

```
  P1   P2   P3   P4
0.67 0.42 0.70 0.62


Proporção de acerto total: 0.61
```

Visto que a análise de discriminante pode ser vista como uma regressão que retorna as coordenadas no espaço que contém os eixos da análise, então tem-se os seguintes coeficientes obtidos pelo ajuste:

```
> adl$scaling
```

```
                     LD1          LD2        LD3
Area_b         1.92656665  0.681760093  0.7486852
Perimetro_b   -0.95271555 -0.002356106 -0.4724736
Comprimento_b -0.08690272  0.286139496 -0.3605042
Largura_b      0.18321143 -0.262576271 -0.7541461
Area_a        -2.18172686  0.358722367  1.0330789
Perimetro_a    0.03561963 -0.795888267 -0.9366602
Comprimento_a  0.59207349  0.916683892  0.1060875
Largura_a      0.65387315 -0.078985960 -0.4565913
```

Além disso, tem-se a constante do ajuste obtida atráves das médias ponderadas pelos pesos das prioris e pelos demais coeficientes ajustados:

```
> # mostra as médias ponderadas tendo as prioris como pesos
> mediapond <- adl$prior %*% adl$means
>
> # mostra a constante da função discriminante
> constante <- mediapond %*% adl$scaling
> constante
```

```
             LD1        LD2          LD3
[1,] 0.004801226 -0.1029199 -0.006367137
```

```
> # mostra os valores singulares (raiz quadrada dos lambdas dos eixos da análise)
> adl$svd
```

```
[1] 9.137368 4.899115 1.402262
```

```
> # mostra a proporção de explicação de cada eixo da análise
> adl$svd^2/sum(adl$svd^2)
```

```
[1] 0.76276390 0.21927196 0.01796414
```

Nota-se que com apenas os eixos LD1 e LD2 da análise discriminante por função linear, tem-se em torno de 98% da representação. Sendo assim, observa-se os dados amostrais preditos pelo modelo no espaço da análise bidimensional com LD1 e LD2.

```
> adl.predito <- predict(adl)
> media_geral_coords <- data.frame(type=c("P1","P2","P3","P4"),lda=(predict(adl,newdata = adl$means))$x)
>
> newdata <-data.frame(type = adl.predito$class, lda = adl.predito$x)
```

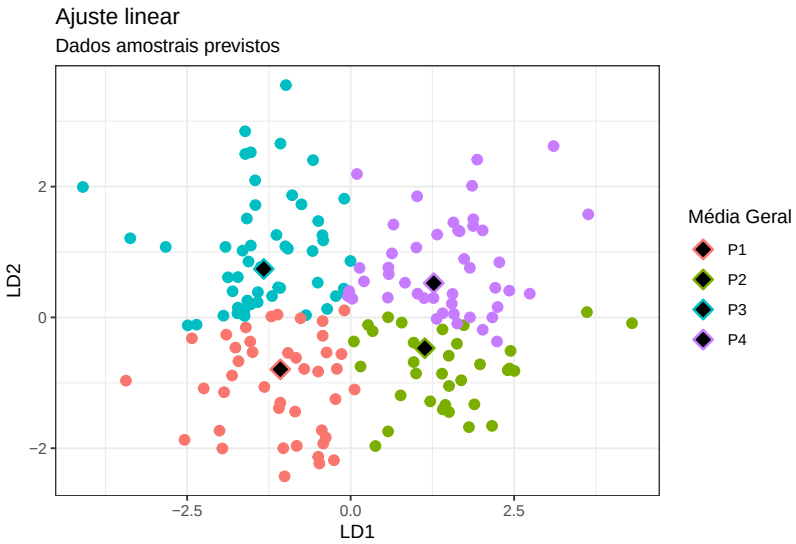


Figura 1: Representação do ajuste linear em 2 dimensões.

Observação: Os pontos são classificados de acordo com a proximidade do ponto médio de cada grupo. Visto que a representação foi feita em 2 dimensões, porém o ajuste foi feito em 3 dimensões, então alguns pontos são mais próximos dos grupos classificados (por distância euclidiana) quando dispostos no espaço 3D, e não necessariamente no espaço 2D.

Função quadrática

Para a análise discriminante via função quadrática, utiliza-se a função `qda(.)` do R. Tal função quadrática é utilizada quando a hipótese de matrizes de covariância iguais entre as populações não é validada.

```
> adq <- qda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1pad[,1], CV=F, prior=rep(1/4,4))
```

Segue com o ajuste em cima dos dados amostrais treinados, da mesma forma que anteriormente.

```
> adq1 <- qda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))
>
> #matriz de confusão
> cvl2<-table(trigo1$Populacao,adq1$class)
```

Tabela 2: Matriz de confusão para o ajuste quadrático.

	P1	**P2**	**P3**	**P4**
P1	20	4	10	2
P2	3	7	2	24
P3	13	2	31	4
P4	0	4	3	43

```
Proporção acerto para cada grupo:


  P1   P2   P3   P4
0.56 0.19 0.62 0.86


Proporção de acerto total: 0.59
```

```
> #função discriminante
> adq$scaling
```

```
, , P1

                   1          2          3          4          5          6          7          8
Area_b        -1.563559 -1.896349 -0.2582576 -0.4785483  1.4383978 -0.0280419 -0.1440780  0.2204616
Perimetro_b    0.000000  1.913804 -1.1844105  1.7914106 -0.2913281  1.6940435 -0.6521358  0.3257151
Comprimento_b  0.000000  0.000000  1.7832539 -1.0103638  0.4850792  0.3715471 -0.8680783  0.7722309
Largura_b      0.000000  0.000000  0.0000000 -1.6511574  0.1278890 -1.3696940  0.4925139 -0.3107585
Area_a         0.000000  0.000000  0.0000000  0.0000000 -1.9899025  1.8944959  0.2304715 -4.0237846
Perimetro_a    0.000000  0.000000  0.0000000  0.0000000  0.0000000 -3.3140599 -1.0679824  0.3232926
Comprimento_a  0.000000  0.000000  0.0000000  0.0000000  0.0000000  0.0000000  2.2205599  0.5166239
Largura_a      0.000000  0.000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  3.3418071

, , P2

                   1          2           3          4           5          6           7           8
Area_b        1.414392  1.939292 -0.03290288 -0.3442230  0.99082530 -0.6732054  0.76820047 -0.70341385
Perimetro_b   0.000000 -1.978863 -0.21959452  0.6832085  0.53956315  1.7753343  0.04431932  0.63929717
Comprimento_b 0.000000  0.000000  0.73645311 -0.7598222 -0.10377980  0.3542642  0.36304414 -0.03349578
Largura_b     0.000000  0.000000  0.00000000  0.9029634 -0.03602697 -0.2286851 -0.25174212  0.07789862
Area_a        0.000000  0.000000  0.00000000  0.0000000 -1.83802164  1.8427190  0.35880783 -1.67205044
Perimetro_a   0.000000  0.000000  0.00000000  0.0000000  0.00000000 -3.2594427  0.06203226 -0.19374698
Comprimento_a 0.000000  0.000000  0.00000000  0.0000000  0.00000000  0.0000000 -1.79136201  0.89592150
Largura_a     0.000000  0.000000  0.00000000  0.0000000  0.00000000  0.0000000  0.00000000  1.20982497

, , P3

                   1          2          3          4          5          6          7          8
Area_b        0.922847  2.004005 -0.6675633  0.50367521  1.04532662  0.8421988  0.9462097 -0.3442682
Perimetro_b   0.000000 -2.406212  1.7357823 -0.06289415  0.77673310  1.2640794 -0.3491507 -0.7382213
Comprimento_b 0.000000  0.000000 -1.9257536 -0.10358830 -0.02850850  1.0135524 -0.7188688 -0.4477824
Largura_b     0.000000  0.000000  0.0000000 -1.33453335 -0.08708544 -0.2084943  0.6028399  0.1362380
Area_a        0.000000  0.000000  0.0000000  0.00000000 -2.13881482  1.3514041  0.3447954  5.1568858
Perimetro_a   0.000000  0.000000  0.0000000  0.00000000  0.00000000 -4.0381977 -2.9592860 -1.5228975
Comprimento_a 0.000000  0.000000  0.0000000  0.00000000  0.00000000  0.0000000  3.0615859 -0.3996643
Largura_a     0.000000  0.000000  0.0000000  0.00000000  0.00000000  0.0000000  0.0000000 -3.3519219

, , P4

                   1         2         3         4         5         6           7         8
Area_b        1.056213 -1.701812  1.038953 -4.369901 -1.5380430 -1.0644488 -0.339005373  0.6509817
Perimetro_b   0.000000  2.173214 -2.401867  1.565423 -0.4852597  1.5046457 -0.727754208 -0.7020955
Comprimento_b 0.000000  0.000000  2.319178  1.143581  0.8347965  0.4215422  0.320684401  0.2389981
Largura_b     0.000000  0.000000  0.000000  6.908581  1.4449508  0.4158411  0.584798112 -0.6389468
Area_a        0.000000  0.000000  0.000000  0.000000  1.7147223  1.3086724  0.004063098 -3.1409645
Perimetro_a   0.000000  0.000000  0.000000  0.000000  0.0000000 -2.4237761  2.356504728  0.3168427
Comprimento_a 0.000000  0.000000  0.000000  0.000000  0.0000000  0.0000000 -2.213021622  1.3909063
Largura_a     0.000000  0.000000  0.000000  0.000000  0.0000000  0.0000000  0.000000000  2.5081673
```

Testando variações no ajuste

Ajuste com os dados originais

Aplicando o mesmo procedimento com os dados originais, chega-se em:

```
> adl1_2 <- lda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))
> adl_2 <- lda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=F, prior=rep(1/4,4))
> cvl1_2<-table(trigo1$Populacao,adl1_2$class)
> cat("Proporção de acerto total:",round(sum(diag(prop.table(cvl1_2))),2))
```

```
Proporção de acerto total: 0.61
```

```
> adq1_2 <- qda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))
> adq_2 <- qda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=F, prior=rep(1/4,4))
> cvq1_2<-table(trigo1$Populacao,adq1_2$class)
> cat("Proporção de acerto total:",round(sum(diag(prop.table(cvq1_2))),2))
```

```
Proporção de acerto total: 0.59
```

Ou seja, a proporção de acerto no ajuste com os dados originais coincide com o ajuste dos dados padronizados. Contudo, nota-se a seguir que os coeficientes das função discriminante mudam:

```
> adl_2$scaling
```

```
                     LD1           LD2          LD3
Area_b         0.007909040  0.0027987965  0.003073541
Perimetro_b   -0.112191329 -0.0002774539 -0.055638263
Comprimento_b -0.015748655  0.0518546722 -0.065331166
Largura_b      0.028526603 -0.0408839629 -0.117422957
Area_a        -0.009059310  0.0014895435  0.004289713
Perimetro_a    0.003826063 -0.0854899004 -0.100610835
Comprimento_a  0.125735185  0.1946707956  0.022529190
Largura_a      0.235096410 -0.0283989568 -0.164164834
```

```
> # mostra as médias ponderadas tendo as prioris como pesos
> mediapond_2 <- adl_2$prior %*% adl_2$means
> # mostra a constante da função discriminante
> constante_2 <- mediapond_2 %*% adl_2$scaling
> constante_2
```

```
          LD1      LD2       LD3
[1,] -5.314927 12.80156 -29.10019
```

```
> #função discriminante
> adq_2$scaling
```

```
, , P1

                      1           2           3           4           5            6            7            8
Area_b        -0.006418804 -0.007784988 -0.001060212 -0.001964561  0.005904984 -0.0001151191 -0.0005914764  0.0009050502
Perimetro_b    0.000000000  0.225368611 -0.139475619  0.210955660 -0.034306656  0.1994897510 -0.0767951966  0.0383560504
Comprimento_b  0.000000000  0.000000000  0.323164221 -0.183099802  0.087906860  0.0673323821 -0.1573145765  0.1399449598
Largura_b      0.000000000  0.000000000  0.000000000 -0.257090478  0.019912729 -0.2132657202  0.0766859804 -0.0483860956
Area_a         0.000000000  0.000000000  0.000000000  0.000000000 -0.008262787  0.0078666241  0.0009570002 -0.0167081917
Perimetro_a    0.000000000  0.000000000  0.000000000  0.000000000  0.000000000 -0.3559779166 -0.1147167459  0.0347263017
Comprimento_a  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.0000000000  0.4715673068  0.1097123983
Largura_a      0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.0000000000  0.0000000000  1.2015279220

, , P2

                      1           2            3           4           5           6           7           8
Area_b        0.005806433  0.007961279 -0.0001350746 -0.001413122  0.004067587 -0.002763677  0.003153656 -0.002887691
Perimetro_b   0.000000000 -0.233029926 -0.0258593460  0.080454311  0.063538699  0.209062520  0.005219022  0.075283330
Comprimento_b 0.000000000  0.000000000  0.1334612491 -0.137696238 -0.018807148  0.064200345  0.065791459 -0.006070160
Largura_b     0.000000000  0.000000000  0.0000000000  0.140594289 -0.005609514 -0.035606996 -0.039197051  0.012129064
Area_a        0.000000000  0.000000000  0.0000000000  0.000000000 -0.007632123  0.007651628  0.001489898 -0.006942951
Perimetro_a   0.000000000  0.000000000  0.0000000000  0.000000000  0.000000000 -0.350111241  0.006663161 -0.020811225
Comprimento_a 0.000000000  0.000000000  0.0000000000  0.000000000  0.000000000  0.000000000 -0.380421072  0.190261608
Largura_a     0.000000000  0.000000000  0.0000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.434985754

, , P3

                      1           2           3           4           5           6           7           8
Area_b        0.003788519  0.008226942 -0.002740515  0.002067713  0.004291328  0.003457438  0.003884429 -0.001413308
Perimetro_b   0.000000000 -0.283354374  0.204404894 -0.007406385  0.091467720  0.148857393 -0.041115820 -0.086932593
Comprimento_b 0.000000000  0.000000000 -0.348988256 -0.018772443 -0.005166358  0.183677640 -0.130274599 -0.081147863
Largura_b     0.000000000  0.000000000  0.000000000 -0.207791099 -0.013559481 -0.032463234  0.093864097  0.021212688
Area_a        0.000000000  0.000000000  0.000000000  0.000000000 -0.008881124  0.005611513  0.001431714  0.021413233
Perimetro_a   0.000000000  0.000000000  0.000000000  0.000000000  0.000000000 -0.433760782 -0.317870077 -0.163581197
Comprimento_a 0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.650171088 -0.084874371
Largura_a     0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000 -1.205164652

, , P4

                      1           2           3          4           5           6            7           8
Area_b        0.004336022 -0.006986367  0.004265161 -0.01793954 -0.006314053 -0.004369830 -1.391702e-03  0.002672444
Perimetro_b   0.000000000  0.255916631 -0.282842724  0.18434341 -0.057143953  0.177186364 -8.569999e-02 -0.082678431
```

```
Comprimento_b 0.000000000  0.000000000  0.420285338  0.20724166  0.151283192  0.076392578  5.811496e-02  0.043311629
Largura_b     0.000000000  0.000000000  0.000000000  1.07568811  0.224983441  0.064747783  9.105493e-02 -0.099486056
Area_a        0.000000000  0.000000000  0.000000000  0.00000000  0.007120140  0.005434076  1.687143e-05 -0.013042407
Perimetro_a   0.000000000  0.000000000  0.000000000  0.00000000  0.000000000 -0.260348575  2.531227e-01  0.034033486
Comprimento_a 0.000000000  0.000000000  0.000000000  0.00000000  0.000000000  0.000000000 -4.699665e-01  0.295378634
Largura_a     0.000000000  0.000000000  0.000000000  0.00000000  0.000000000  0.000000000  0.000000e+00  0.901797407
```

Ajuste com prioris amostrais

Para utilizar prioris amostrais, suspende-se o uso do argumento `prior` na funções `lda(.)` e `qda(.)` usadas no R.

LDA:

```
> # lda com priori amostral com dados originais
> adl1_3 <- lda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T)
> cvl1_3<-table(trigo1$Populacao,adl1_3$class)
> cat("Proporção de acerto total:",round(sum(diag(prop.table(cvl1_3))),2))
```

```
Proporção de acerto total: 0.63
```

QDA:

```
> # qda com priori amostral com dados originais
> adq1_3 <- qda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T)
> cvq1_3<-table(trigo1$Populacao,adq1_3$class)
> cat("Proporção de acerto total:",round(sum(diag(prop.table(cvq1_3))),2))
```

```
Proporção de acerto total: 0.58
```

Portanto percebe-se, que convém utilzar o ajuste com prioris não informativa.

Decisão

Primeiro pode-se destacar que a proporção de acerto da previsão com os dados padronizados e prioris não informativas do próprio ajuste linear foi de 61%, enquanto que para o ajuste quadrático foi de 59%. Contudo, vale testar se as matrizes de covariâncias de cada grupo são diferentes ou não para definir qual função usar.

Observa-se, a seguir, as matrizes de covariância amostrais (arredondadas para 2 casas decimais). Nota-se que os dados considerados são referentes aos dados originais padronizados.

P1:

	Area_b	Perimetro_b	Comprimento_b	Largura_b	Area_a	Perimetro_a	Comprimento_a	Largura_a
Area_b	0.41	0.41	0.33	0.12	0.32	0.38	0.39	0.16
Perimetro_b	0.41	0.67	0.51	0.30	0.34	0.47	0.54	0.10
Comprimento_b	0.33	0.51	0.70	0.03	0.34	0.52	0.65	0.02
Largura_b	0.12	0.30	0.03	0.65	0.09	-0.06	-0.07	0.14
Area_a	0.32	0.34	0.34	0.09	0.52	0.47	0.40	0.40
Perimetro_a	0.38	0.47	0.52	-0.06	0.47	0.68	0.65	0.21
Comprimento_a	0.39	0.54	0.65	-0.07	0.40	0.65	0.93	0.04
Largura_a	0.16	0.10	0.02	0.14	0.40	0.21	0.04	0.53

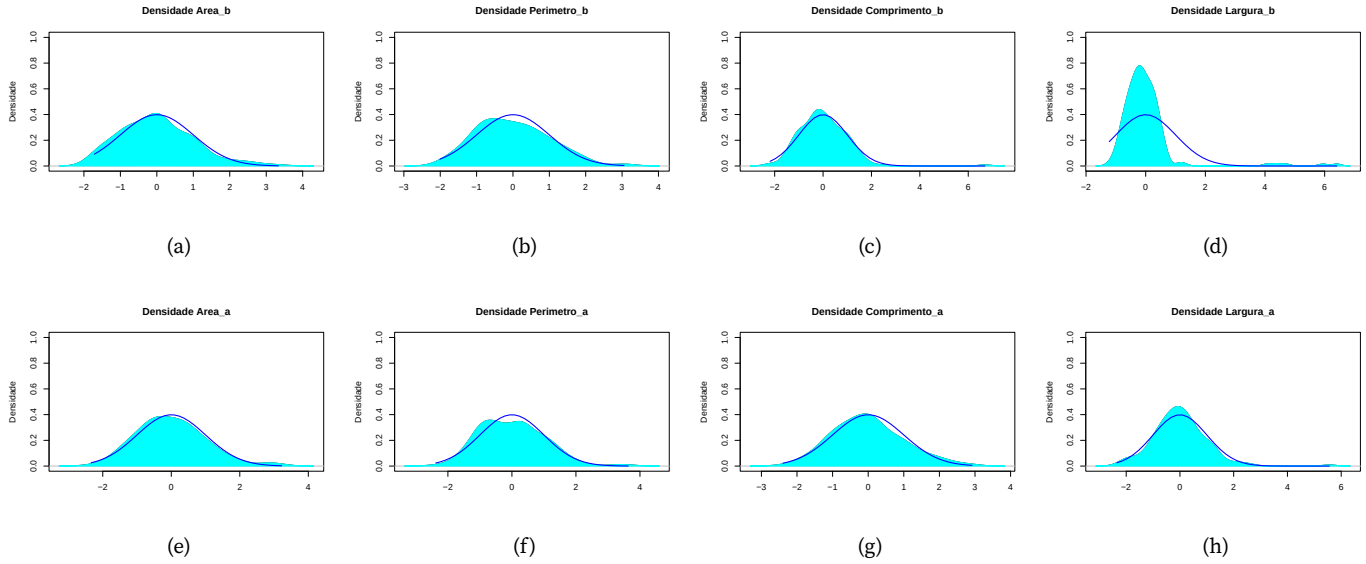
P2:

	Area_b	Perimetro_b	Comprimento_b	Largura_b	Area_a	Perimetro_a	Comprimento_a	Largura_a
Area_b	0.50	0.49	0.17	-0.04	0.40	0.41	0.36	0.40
Perimetro_b	0.49	0.74	0.24	-0.17	0.47	0.60	0.42	0.35
Comprimento_b	0.17	0.24	1.92	1.50	0.02	0.21	0.27	-0.21
Largura_b	-0.04	-0.17	1.50	2.60	-0.21	-0.22	-0.13	-0.28
Area_a	0.40	0.47	0.02	-0.21	0.65	0.56	0.37	0.72
Perimetro_a	0.41	0.60	0.21	-0.22	0.56	0.69	0.40	0.53
Comprimento_a	0.36	0.42	0.27	-0.13	0.37	0.40	0.64	0.11
Largura_a	0.40	0.35	-0.21	-0.28	0.72	0.53	0.11	1.74

```
P3:

              Area_b Perimetro_b Comprimento_b Largura_b Area_a Perimetro_a Comprimento_a Largura_a
Area_b          1.17        0.98          0.47      0.36   0.91        0.96          0.61      0.51
Perimetro_b     0.98        0.99          0.55      0.28   0.82        0.91          0.67      0.38
Comprimento_b   0.47        0.55          0.60      0.11   0.42        0.56          0.53      0.08
Largura_b       0.36        0.28          0.11      0.68   0.25        0.24          0.01      0.19
Area_a          0.91        0.82          0.42      0.25   0.94        0.85          0.58      0.68
Perimetro_a     0.96        0.91          0.56      0.24   0.85        0.96          0.72      0.43
Comprimento_a   0.61        0.67          0.53      0.01   0.58        0.72          0.75      0.20
Largura_a       0.51        0.38          0.08      0.19   0.68        0.43          0.20      0.77

P4:

              Area_b Perimetro_b Comprimento_b Largura_b Area_a Perimetro_a Comprimento_a Largura_a
Area_b          0.90        0.70          0.33      0.35   0.55        0.45          0.26      0.51
Perimetro_b     0.70        0.76          0.47      0.19   0.45        0.52          0.32      0.36
Comprimento_b   0.33        0.47          0.53      0.01   0.16        0.33          0.23      0.03
Largura_b       0.35        0.19          0.01      0.20   0.20        0.11          0.05      0.22
Area_a          0.55        0.45          0.16      0.20   0.71        0.49          0.36      0.65
Perimetro_a     0.45        0.52          0.33      0.11   0.49        0.64          0.51      0.27
Comprimento_a   0.26        0.32          0.23      0.05   0.36        0.51          0.65      0.04
Largura_a       0.51        0.36          0.03      0.22   0.65        0.27          0.04      0.94
```

Procede-se então com o teste M de Box para testar a hipótese nula de homogeneidade das matrizes de covariância. Utiliza-se então a função `boxM(.)` da biblioteca `biotools` do software R. Nessa função, atribui-se dois argumentos: a matriz de variáveis quantitativas que busca-se comparar e o vetor que identifica os grupos a serem comparados.

```
> biotools::boxM(trigo1pad[,-1], trigo1pad[,1])


    Box's M-test for Homogeneity of Covariance Matrices

data:  trigo1pad[, -1]
Chi-Sq (approx.) = 458.91, df = 108, p-value < 2.2e-16
```



(a) (b) (c) (d)

(e) (f) (g) (h)

Figura 2: Verificando normalidade multivarida dos dados.

Como o teste Box-M é sensível a normalidade, pode-se ter que um resultado significativo seja devido à não normalidade dos dados do que à matrizes de covariância não homogêneas. Percebe-se pelo gráfico que a hipótese de normalidade não parece estar adequada. Dessa forma, o resultado de não aceitação da hipótese nula do teste (apresentou p-valor < 0,05) pode não ser coerente.

Conclui-se então que, pela proporção de acerto do primeiro ajuste que para a previsão dos novos dados amostrais, utiliza-se a função linear. Além disso, nota-se a seguinte proporção de acerto por população em cada ajuste:

```
Proporção de acerto para cada grupo (função linear):

  P1   P2   P3   P4
0.67 0.42 0.70 0.62

Proporção acerto para cada grupo (função quadrática):

  P1   P2   P3   P4
0.56 0.19 0.62 0.86
```

Ou seja, a função linear apresentou proporção de acerto mais balanceado por grupo comparado com a função quadrática. A função quadrática acaba acertando mais que a função linear apenas para a população 4 (P4).

Predição

A seguir, será realizada a previsão dos novos dados amostrais através da análise de discriminante por função linear:

Tabela 3: Novas observações para previsão - dados originais - 'novostrigo.xls'.

Rotulo	Area_b	Perimetro_b	Comprimento_b	Largura_b	Area_a	Perimetro_a	Comprimento_a	Largura_a
1	2697	219	86	45	2775	228	97	50
2	2835	221	82	46	2739	216	94	49
3	3005	223	84	47	3080	224	90	47
4	2791	212	88	86	2639	242	95	43
5	2599	207	92	47	2428	226	99	43
6	3074	203	91	44	2343	220	96	42
7	2785	208	85	47	3007	231	90	43
8	3009	218	89	44	2462	223	90	42
9	2603	211	85	46	2604	238	92	43
10	2874	221	94	44	2865	225	91	45
11	2706	204	90	44	2715	235	87	37

Usa-se portanto os novos dados originais no ajuste dos dados inalterados, e tem-se a seguinte representação em duas dimensões (Nota-se que há influência de uma terceira dimensão).

```
> adl.predito <- predict(adl_2,newdata = novos)
> media_geral_coords <- data.frame(type=c("P1","P2","P3","P4"),lda=(predict(adl_2,newdata = adl_2$means))$x)
>
> newdata <-data.frame(type = adl.predito$class, lda = adl.predito$x)
```

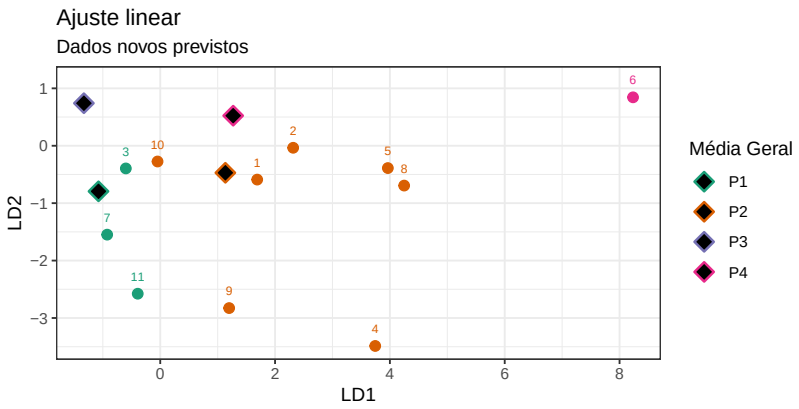


Figura 3: Predição das novas observações em 2 dimensões das funções discriminantes de Fisher.

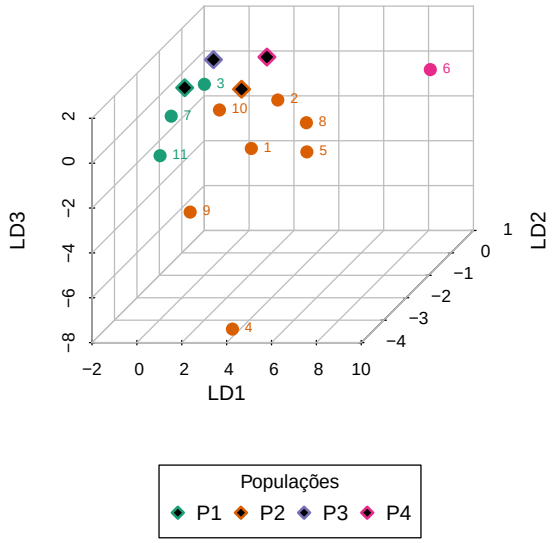


Figura 4: Predição das novas observações no espaço máximo das funções discriminantes de Fisher.

Tabela 4: Resultado da predição.

População	Rótulos
P1	3,7,11
P2	1,2,4,5,8,9,10
P3	
P4	6

Tabela 5: Novas observações com cor da previsão - dados originais - 'novostrigo.xls'.

Rotulo	Area_b	Perimetro_b	Comprimento_b	Largura_b	Area_a	Perimetro_a	Comprimento_a	Largura_a
1	2697	219	86	45	2775	228	97	50
2	2835	221	82	46	2739	216	94	49
3	3005	223	84	47	3080	224	90	47
4	2791	212	88	86	2639	242	95	43
5	2599	207	92	47	2428	226	99	43
6	3074	203	91	44	2343	220	96	42
7	2785	208	85	47	3007	231	90	43
8	3009	218	89	44	2462	223	90	42
9	2603	211	85	46	2604	238	92	43
10	2874	221	94	44	2865	225	91	45
11	2706	204	90	44	2715	235	87	37

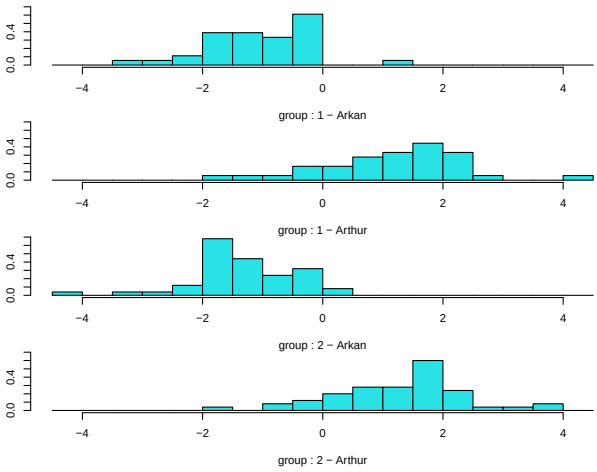
Comentário: Disso então tivemos que 7 de 11 dos novos trigos preditos pelo modelo foram classificados na categoria P2, 3 foram classificados na categoria P1 e um foi classificado na categporia P4, além de que nenhum fora classificado na categoria P3. Válido dizer que apesar de ter-se predito majoritariamente os pontos na categoria P2 vimos, ao avaliar a matriz de *cross-validation*, que esta categoria foi a com menor percentual de acertos, o que pode indicar uma desconfiança dos resultados.

Interpretação das variáveis resposta

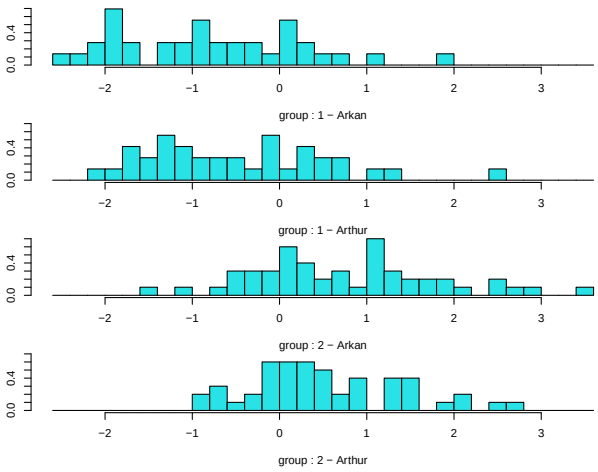
Nota-se que, para verificar qual das duas variáveis (Localidade ou Variedade) é mais importante para classificar uma nova unidade amostral, avalia-se o histograma das funções discriminantes de Fisher:

```
> par(mai=c(0.1,1,0.1,1), mar=c(4,3,0,3))
> ldahist((predict(adl_2))$x[,1], g=nomeacao)
```

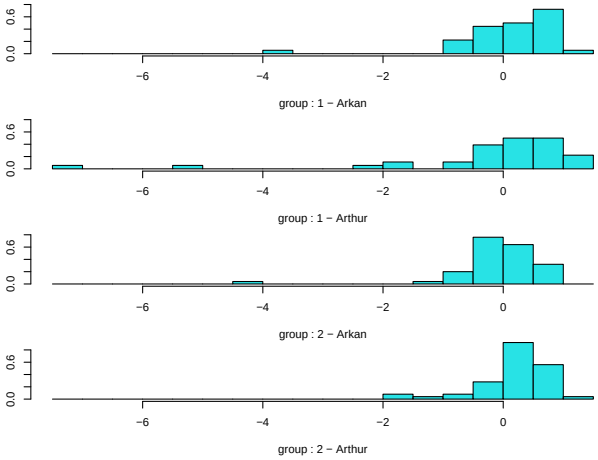
```
>
> par(mai=c(0.1,1,0.1,1), mar=c(4,3,0,3))
> ldahist((predict(adl_2))$x[,2], g=nomeacao)
>
> par(mai=c(0.1,1,0.1,1), mar=c(4,3,0,3))
> ldahist((predict(adl_2))$x[,3], g=nomeacao)
```



(a) LD1



(b) LD2



(c) LD3

Figura 5: Histrograma das funções discriminante da função linear.

Proporção do traço para o ajuste com os dados originais:

```
LD1: 0.7627639   LD2: 0.219272   LD3: 0.01796414
```

Pelo traço do ajuste, pode-se ver que o primeiro eixo da função linear corresponde a 76,3%, enquanto que LD2 representa 21,9%, e LD3, 0,02%. Sendo assim, pode-se analisar o histrograma das funções discriminantes somente de LD1 e percebe-se que os grupos P2 e P4 possuem uma grande área de confundimento, assim como os grupos P1 e P3 se apresentam com uma grande área de confundimento, porém no lado oposto (abaixo do ponto 0 no eixo da abscissa).

Sendo assim, a variável Localidade é a que melhor discrimina os grupos, pois apresenta uma pequena área de interseção comparado com o histograma da mesma localidade porém com Variedade diferente. Logo, Variedade é menos importante para discriminar, pois não diferencia tanto quanto Localidade.

Código

```
library(showtext)
library(tidyverse)
library(dplyr)
library(kableExtra)
library(knitr)
library(readxl)
library(latex2exp)
library(ggplot2)
library(MASS)

# Lendo o arquivo
trigo <- read_excel("trigo.xls")

# Atribuindo estrutura fatorial para as colunas das variáveis respostas: localidade e variedade
trigo$Localidade <- as.factor(trigo$Localidade)
trigo$Variedade <- as.factor(trigo$Variedade)

# Transformando em dataframe para realizar a análise
trigo <- as.data.frame(trigo)

# Criando a coluna de população referente à junção das variáveis respostas
trigo1 <- trigo%>%mutate(Populacao=case_when(
 (Localidade == 1 & Variedade == "Arkan") ~ "P1",
 (Localidade == 1 & Variedade == "Arthur") ~ "P2",
 (Localidade == 2 & Variedade == "Arkan") ~ "P3",
 (Localidade == 2 & Variedade == "Arthur") ~ "P4"
))%>%dplyr::select(-Localidade,-Variedade)%>%dplyr::select(Populacao,everything())

# mostrando estrutura dos dados no R
str(trigo)

# mádia amostral (para cada variável preditora)
media_trigo <- sapply(trigo1%>%dplyr::select(-Populacao), mean)
media_trigo

# desvio padrão amostral (para cada variável preditora)
dp_trigo <- sapply(trigo1%>%dplyr::select(-Populacao), sd)
dp_trigo

trigo1pad <- cbind(trigo1%>%dplyr::select(Populacao),scale(trigo1%>%dplyr::select(-Populacao)))

# tabela de contigência das variáveis Localidade e Variedade da amostra para ajuste
table(trigo[,1:2])

# priori amostral
round(table(trigo1pad$Populacao)/nrow(trigo1pad),2)

adl <- lda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1pad[,1], CV=F, prior=rep(1/4,4))

adl1 <- lda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))

#matriz de confusão
cvl1<-table(trigo1$Populacao,adl1$class)

m_confund <- pivot_wider(data.frame(cvl1) ,names_from = Var2, values_from = Freq)
colnames(m_confund)[1] <- ""

kableExtra::kbl(m_confund,
   booktabs = F,
   vline="",
   linesep="",
   caption="Matriz de confusão para o ajuste linear."
   )%>%
 kableExtra::column_spec(1,bold = T)%>%kableExtra::row_spec(0,bold=T)%>%
 kableExtra::kable_styling(latex_options = c("HOLD_position"), position="center")

cat("Proporção de acerto para cada grupo:")
round(diag(prop.table(cvl1,1)),2)

cat("Proporção de acerto total:",round(sum(diag(prop.table(cvl1))),2))

adl$scaling
```

```
# mostra as médias ponderadas tendo as prioris como pesos
mediapond <- adl$prior %*% adl$means

# mostra a constante da função discriminante
constante <- mediapond %*% adl$scaling
constante

# mostra os valores singulares (raiz quadrada dos lambdas dos eixos da análise)
adl$svd

# mostra a proporção de explicação de cada eixo da análise
adl$svd^2/sum(adl$svd^2)

adl.predito <- predict(adl)
media_geral_coords <- data.frame(type=c("P1","P2","P3","P4"),lda=(predict(adl,newdata = adl$means))$x)

newdata <-data.frame(type = adl.predito$class, lda = adl.predito$x)

ggplot(newdata) + geom_point(aes(lda.LD1, lda.LD2, colour = type), size = 2.5) +
 geom_point(aes(lda.LD1,lda.LD2,colour=type),data = media_geral_coords, size=3.5, shape=23,fill="black",stroke=1)+
 coord_fixed(ratio = 1)+
 labs(x="LD1",y="LD2", title = "Ajuste linear",subtitle="Dados amostrais previstos", colour="Média Geral")+
 theme_bw()

adq <- qda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1pad[,1], CV=F, prior=rep(1/4,4))

adq1 <- qda(trigo1pad[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))

#matriz de confusão
cvl2<-table(trigo1$Populacao,adq1$class)

m_confund2 <- pivot_wider(data.frame(cvl2) ,names_from = Var2, values_from = Freq)
colnames(m_confund2)[1] <- ""

kableExtra::kbl(m_confund2,
   booktabs = F,
   vline="",
   linesep="",
   caption="Matriz de confusão para o ajuste quadrático."
   )%>%
 kableExtra::column_spec(1,bold = T)%>%kableExtra::row_spec(0,bold=T)%>%
 kableExtra::kable_styling(latex_options = c("HOLD_position"), position="center")

cat("Proporção acerto para cada grupo:")
round(diag(prop.table(cvl2,1)),2)

cat("Proporção de acerto total:",round(sum(diag(prop.table(cvl2))),2))

#função discriminante
adq$scaling

adl1_2 <- lda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))
adl_2 <- lda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=F, prior=rep(1/4,4))
cvl1_2<-table(trigo1$Populacao,adl1_2$class)
cat("Proporção de acerto total:",round(sum(diag(prop.table(cvl1_2))),2))

adq1_2 <- qda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T, prior=rep(1/4,4))
adq_2 <- qda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=F, prior=rep(1/4,4))
cvq1_2<-table(trigo1$Populacao,adq1_2$class)
cat("Proporção de acerto total:",round(sum(diag(prop.table(cvq1_2))),2))

adl_2$scaling
# mostra as médias ponderadas tendo as prioris como pesos
mediapond_2 <- adl_2$prior %*% adl_2$means
# mostra a constante da função discriminante
constante_2 <- mediapond_2 %*% adl_2$scaling
constante_2

#função discriminante
adq_2$scaling

# lda com priori amostral com dados originais
adl1_3 <- lda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T)
cvl1_3<-table(trigo1$Populacao,adl1_3$class)
cat("Proporção de acerto total:",round(sum(diag(prop.table(cvl1_3))),2))
```

```
# qda com priori amostral com dados originais
adq1_3 <- qda(trigo1[,2:ncol(trigo1)], grouping= trigo1[,1], CV=T)
cvq1_3<-table(trigo1$Populacao,adq1_3$class)
cat("Proporção de acerto total:",round(sum(diag(prop.table(cvq1_3))),2))

cat("P1:")
round(cov(trigo1pad%>%dplyr::filter(Populacao=="P1")%>%dplyr::select(-Populacao)),2)

cat("P2:")
round(cov(trigo1pad%>%dplyr::filter(Populacao=="P2")%>%dplyr::select(-Populacao)),2)

cat("P3:")
round(cov(trigo1pad%>%dplyr::filter(Populacao=="P3")%>%dplyr::select(-Populacao)),2)

cat("P4:")
round(cov(trigo1pad%>%dplyr::filter(Populacao=="P4")%>%dplyr::select(-Populacao)),2)

biotools::boxM(trigo1pad[,-1], trigo1pad[,1])

# normalidade <- MVN::mvn(data=trigo1pad[,-1],mvnTest="mardia",multivariatePlot="qq")
for (i in 2:ncol(trigo1pad)){
 x <- trigo1pad[,i]
 var <- colnames(trigo1pad)[i]
 d <- density(x)
 plot(d, main=paste0("Densidade ", var), xlab="", ylab="Densidade", ylim=c(0,1))
 polygon(d, col="cyan", border="cyan")
 xfit<-seq(min(x),max(x),length=40)
 yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
 lines(xfit, yfit, col="blue", lwd=2)
}

cat("Proporção de acerto para cada grupo (função linear):")
round(diag(prop.table(cvl1,1)),2)
cat("Proporção acerto para cada grupo (função quadrática):")
round(diag(prop.table(cvl2,1)),2)

novos <- read_excel("novostrigo.xls")

cbind(`Rotulo`=1:nrow(novos),novos)%>%
 kbl(booktabs = F,
   vline="",
   linesep="",
   caption="Novas observações para previsão - dados originais - `novostrigo.xls`."
   )%>%
 kableExtra::kable_styling(latex_options = c("HOLD_position"), position="center", font_size = 9)

adl.predito <- predict(adl_2,newdata = novos)
media_geral_coords <- data.frame(type=c("P1","P2","P3","P4"),lda=(predict(adl_2,newdata = adl_2$means))$x)

newdata <-data.frame(type = adl.predito$class, lda = adl.predito$x)

ggplot(newdata) +
 geom_point(aes(lda.LD1, lda.LD2, colour = type), size = 2.5) +
 geom_text(aes(lda.LD1, lda.LD2, colour = type, label=1:11),position = position_nudge(y=0.3), size=2.5)+
 geom_point(aes(lda.LD1,lda.LD2,colour=type),data = media_geral_coords, size=3.5, shape=23,fill="black",stroke=1)+
 coord_fixed(ratio = 1)+
 labs(x="LD1",y="LD2", title = "Ajuste linear",subtitle="Dados novos previstos", colour="Média Geral")+
 theme_bw()+
 scale_color_brewer(palette = "Dark2")

library(plotly)

media_geral_coords$type <- c("P11","P22","P33","P44")
newdata

a <- rbind(media_geral_coords,newdata)

plot_ly(x=a$lda.LD1, y=a$lda.LD2, z=a$lda.LD3, type="scatter3d", mode="markers", color=a$type,size = 2)
plot_ly(x=media_geral_coords$lda.LD1, y=media_geral_coords$lda.LD2, z=media_geral_coords$lda.LD3, type="scatter3d", mode="markers", color=media_g

#http://www.sthda.com/english/wiki/scatterplot3d-3d-graphics-r-software-and-data-visualization
library(scatterplot3d)

# 1. Source the function
source('http://www.sthda.com/sthda/RDoc/functions/addgrids3d.r')
angulo <- 45
```

```
# # 2. Empty 3D scatter plot using pch=""
comb <- rbind(media_geral_coords,newdata)
s3d <- scatterplot3d(comb[,2:4],pch = "",box = F, grid = F, xlab="LD1",ylab="LD2", zlab="LD3", angle=angulo,asp = 1,mar = c(8,4,0,4))
# # 3. Add grids
addgrids3d(comb[,2:4], grid = c("xy", "xz", "yz"), angle = angulo)
# # 4. Add points
colors_ref <- RColorBrewer::brewer.pal(4,"Dark2")
colors <- colors_ref[as.numeric(newdata$type)]
s3d$points3d(newdata[,2:4], pch = 16, col = colors,cex=1.4)
colors2 <- colors_ref[as.numeric(as.factor(media_geral_coords$type))]
s3d$points3d(media_geral_coords[,2:4], pch = 23, col = colors2,cex=1.4, lwd=2, bg="black")
text(s3d$xyz.convert(newdata[,2:4]), labels = 1:11,
    cex= 0.7, col = colors,pos = 4)

legend("bottom", legend = levels(newdata$type),
      pch = c(23,23,23,23), pt.bg  = "black", col = colors2, pt.lwd = 2,
     inset = -0.5, xpd = TRUE, horiz = TRUE,title = "Populações",title.cex = 0.9)

as.data.frame(cbind(`População`=levels(adl.predito$class),`Rótulos`=c("3,7,11","1,2,4,5,8,9,10","","6")))%>%
 kbl(booktabs = F,
     align="c|l",
     vline="",
     linesep="",
     caption="Resultado da predição."
   )%>%
 kableExtra::kable_styling(latex_options = c("HOLD_position"), position="center", font_size = 11)%>%
 row_spec(1,color = colors_ref[1])%>%
 row_spec(2,color = colors_ref[2])%>%
 row_spec(3,color = colors_ref[3])%>%
 row_spec(4,color = colors_ref[4])%>%
 row_spec(0,bold=T,font_size = 9,align = "c")

cbind(`Rotulo`=1:nrow(novos),novos)%>%
 kbl(booktabs = F,
   vline="",
   linesep="",
   align="c|rrrrrrrr",
   caption="Novas observações com cor da previsão - dados originais - `novostrigo.xls`."
   )%>%
 kableExtra::kable_styling(latex_options = c("HOLD_position"), position="center", font_size = 9)%>%
 row_spec(c(3,7,11), background = "#81FFBE")%>%
 row_spec(c(1,2,4,5,8,9,10), background = "#F5A852")%>%
 row_spec(c(6), background = "#E7898A")%>%
 row_spec(0,bold=T,font_size = 8,align = "c")

nomeacao <- (trigo1%>%mutate(nomeacao=case_when(
 Populacao =="P1" ~ ": 1 - Arkan",
 Populacao =="P2" ~ ": 1 - Arthur",
 Populacao =="P3" ~ ": 2 - Arkan",
 Populacao =="P4" ~ ": 2 - Arthur"
))%>%dplyr::select(nomeacao))[,1]

par(mai=c(0.1,1,0.1,1), mar=c(4,3,0,3))
ldahist((predict(adl_2))$x[,1], g=nomeacao)

par(mai=c(0.1,1,0.1,1), mar=c(4,3,0,3))
ldahist((predict(adl_2))$x[,2], g=nomeacao)

par(mai=c(0.1,1,0.1,1), mar=c(4,3,0,3))
ldahist((predict(adl_2))$x[,3], g=nomeacao)

cat("LD1:",(adl_2$svd^2/sum(adl_2$svd^2))[1], "\t","LD2:",(adl_2$svd^2/sum(adl_2$svd^2))[2],"\t","LD3:",(adl_2$svd^2/sum(adl_2$svd^2))[3])
```