

Tarefa2

Lucas Gerlach Nachtigall

16/09/2022

R Markdown

```
library(caret)

## Carregando pacotes exigidos: ggplot2

## Carregando pacotes exigidos: lattice

library(ggplot2)
library(gridExtra)

## Warning: package 'gridExtra' was built under R version 4.2.1

library(GGally)

## Warning: package 'GGally' was built under R version 4.2.1

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v tibble  3.1.7      v dplyr    1.0.9
## v tidyr   1.2.0      v stringr  1.4.0
## v readr   2.1.2      vforcats  0.5.1
## v purrr   0.3.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
```

```

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.2.1

## corrplot 0.92 loaded

```

1 Carregue a base de dados e mostre a estrutura do dataset (str()). O arquivo do dataset não pode ser modificado de forma alguma. A leitura deverá tratar qualquer característica do arquivo.

```

rm(list=ls())
setwd("~/Ensino-Entreterimento/Graduação UCS/Semestre 8/Computação Aplicada I/Tarefa2-Computação_Aplicada")
dadosMain = read.csv2("Dry_Bean_Dataset.csv",header=T)
dados = dadosMain
str(dados)

```

```

## 'data.frame':    13611 obs. of  17 variables:
##   $ Area          : int  28395 28734 29380 30008 30140 30279 30477 30519 30685 30834 ...
##   $ Perimeter     : num  610 638 624 646 620 ...
##   $ MajorAxisLength: num  208 201 213 211 202 ...
##   $ MinorAxisLength: num  174 183 176 183 190 ...
##   $ AspectRatio    : num  1.2 1.1 1.21 1.15 1.06 ...
##   $ Eccentricity   : num  0.55 0.412 0.563 0.499 0.334 ...
##   $ ConvexArea     : int  28715 29172 29690 30724 30417 30600 30970 30847 31044 31120 ...
##   $ EquivDiameter  : num  190 191 193 195 196 ...
##   $ Extent         : num  0.764 0.784 0.778 0.783 0.773 ...
##   $ Solidity       : num  0.989 0.985 0.99 0.977 0.991 ...
##   $ roundness      : num  0.958 0.887 0.948 0.904 0.985 ...
##   $ Compactness     : num  0.913 0.954 0.909 0.928 0.971 ...
##   $ ShapeFactor1   : num  0.00733 0.00698 0.00724 0.00702 0.0067 ...
##   $ ShapeFactor2   : num  0.00315 0.00356 0.00305 0.00321 0.00366 ...
##   $ ShapeFactor3   : num  0.834 0.91 0.826 0.862 0.942 ...
##   $ ShapeFactor4   : num  0.999 0.998 0.999 0.994 0.999 ...
##   $ Class          : chr  "SEKER" "SEKER" "SEKER" "SEKER" ...

```

```
summary(iris)
```

```

##   Sepal.Length   Sepal.Width    Petal.Length   Petal.Width
##   Min.   :4.300  Min.   :2.000  Min.   :1.000  Min.   :0.100
##   1st Qu.:5.100  1st Qu.:2.800  1st Qu.:1.600  1st Qu.:0.300
##   Median :5.800  Median :3.000  Median :4.350  Median :1.300
##   Mean   :5.843  Mean   :3.057  Mean   :5.014  Mean   :1.587
##   3rd Qu.:6.400  3rd Qu.:3.300  3rd Qu.:5.100  3rd Qu.:1.800
##   Max.   :7.900  Max.   :4.400  Max.   :6.900  Max.   :2.500
## 
##   Species
##   setosa    :50
##   versicolor:50
##   virginica :50
## 
## 
## 

```

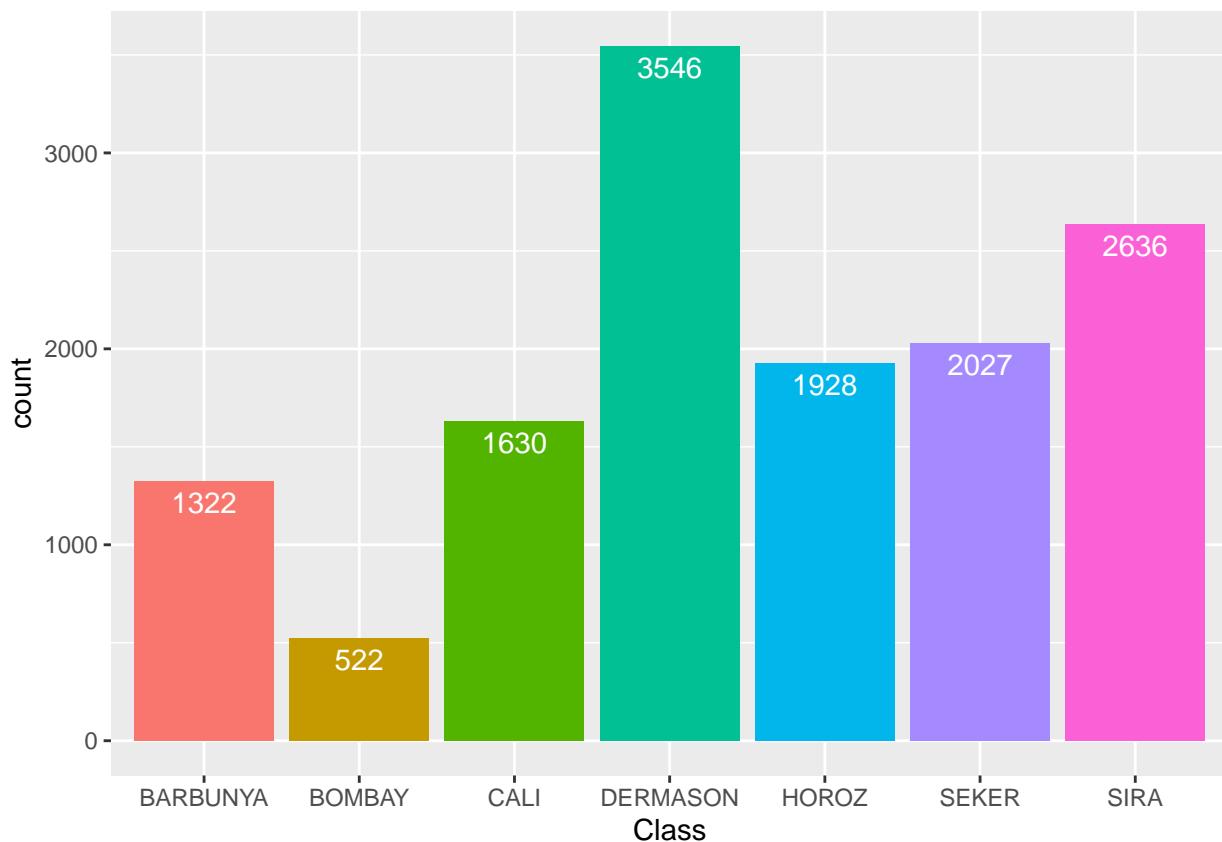
2 Altere a variável do tipo do feijão (Class) para um factor.

```
dados$Class <- factor(dados$Class)
str(dados$Class)
```

```
## Factor w/ 7 levels "BARBUNYA","BOMBAY",...: 6 6 6 6 6 6 6 6 6 ...
```

3 Plote um gráfico de barras que ilustre as quantidades de cada classe.

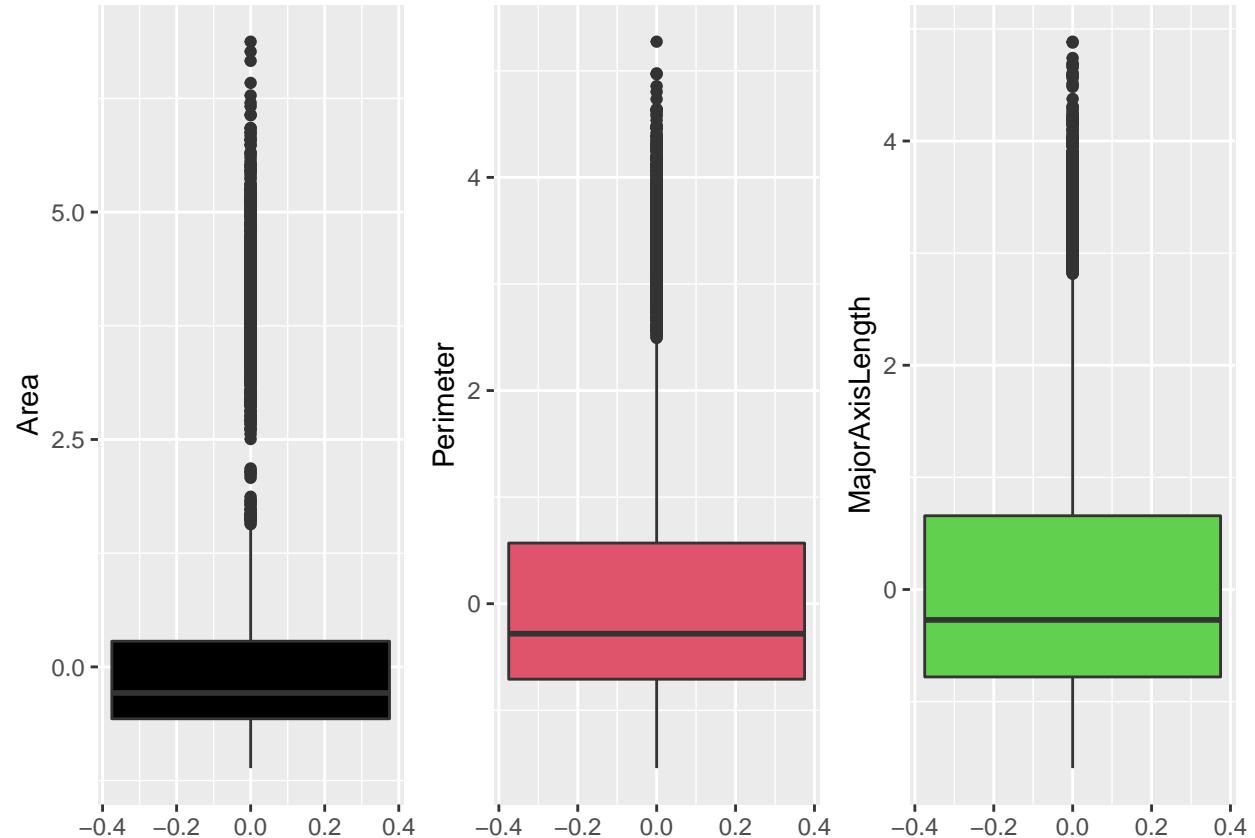
```
ggplot(dados,aes(Class,fill=Class)) + geom_bar() + geom_text(aes(label = ..count..), stat = "count", vj
```

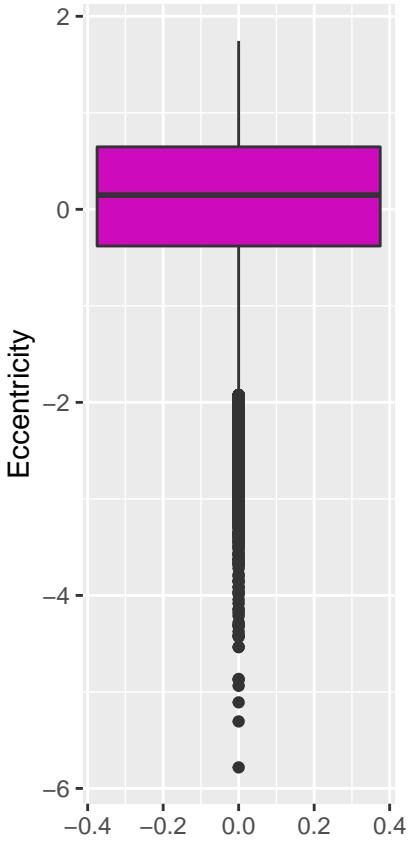
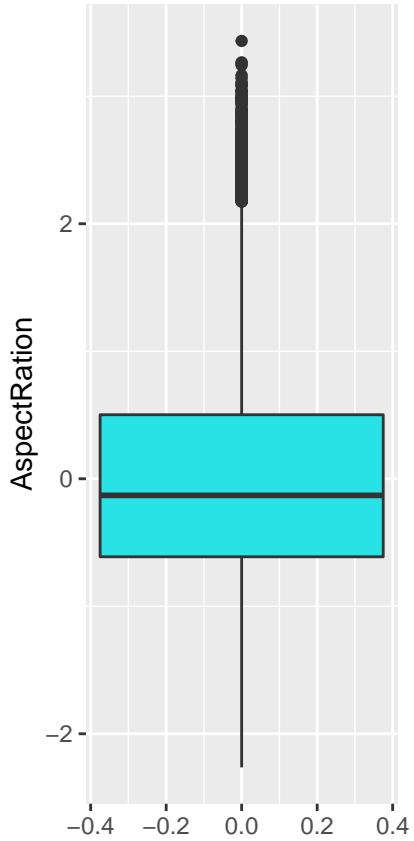
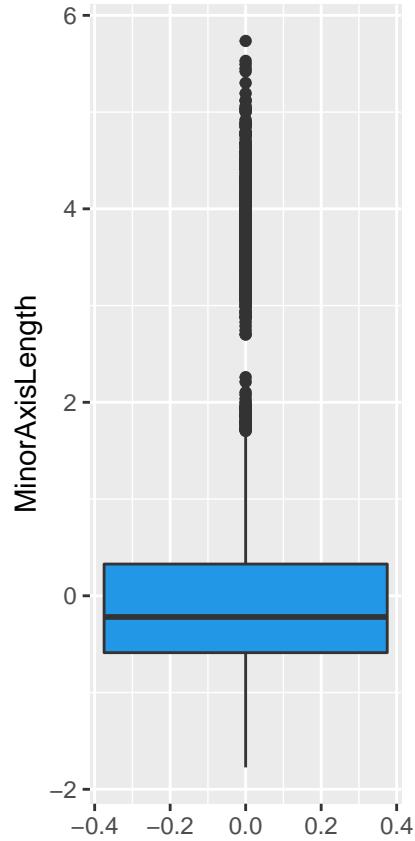


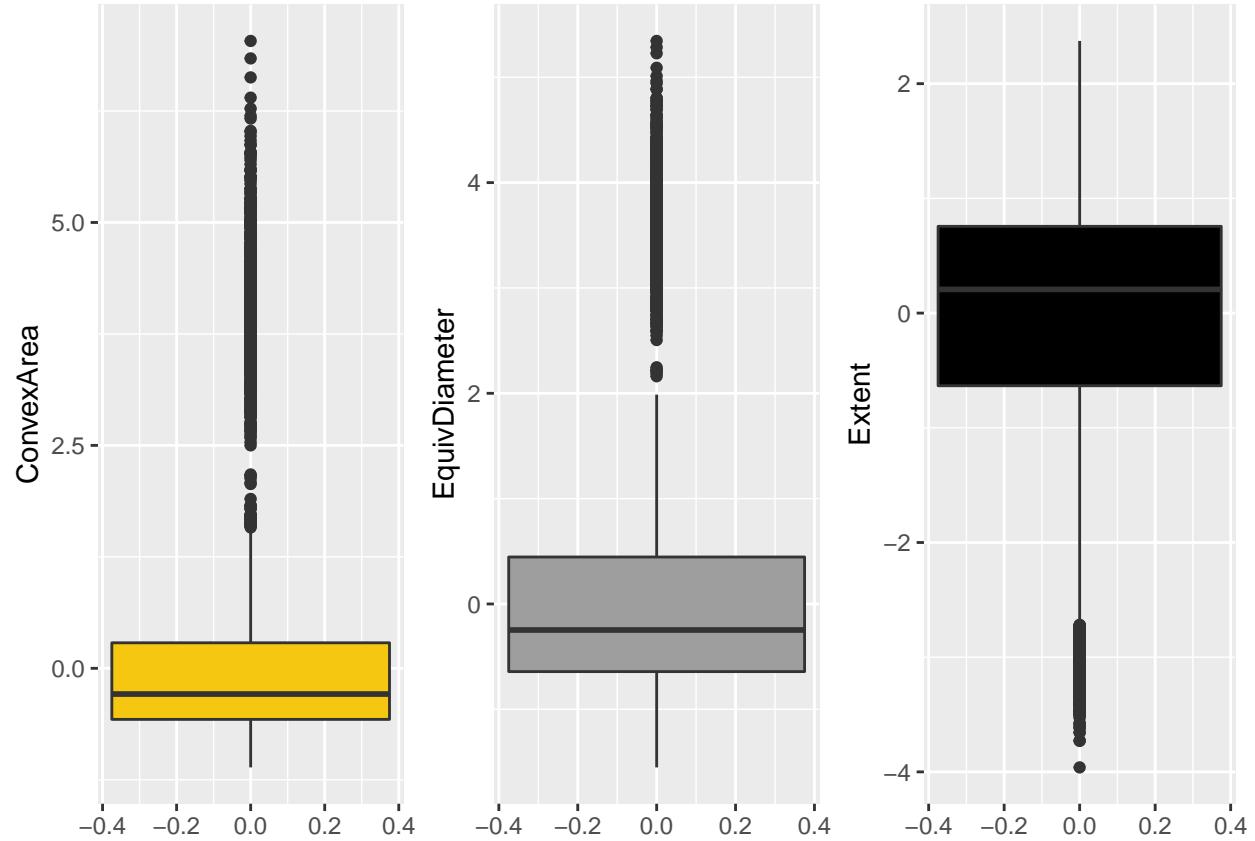
4 Realize a normalização dos dados via Z-score. Plote um boxplot para ilustrar a distribuição de cada variável. Mostre as estatísticas de cada variável (summary).

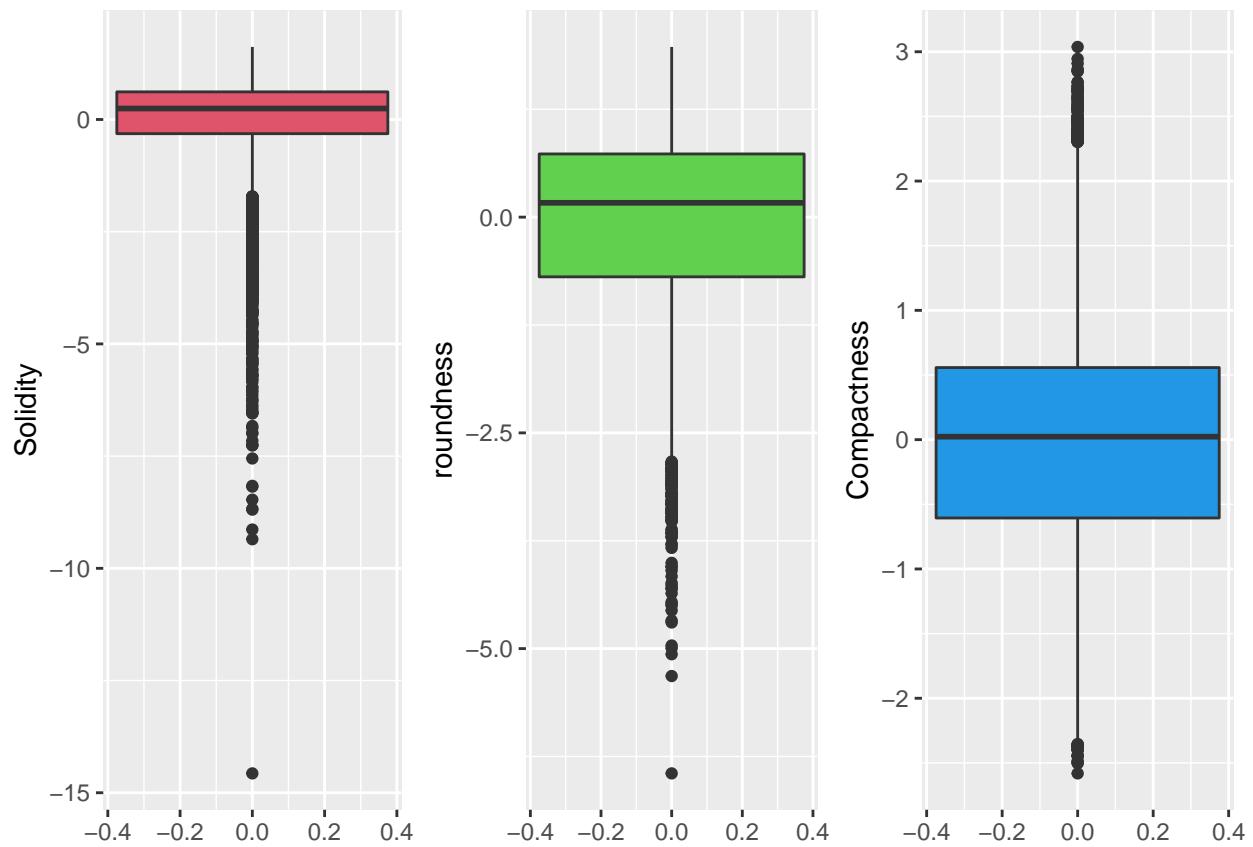
```
normalizacaoParametros = preProcess(dados,method = c("center","scale"))
dados = predict(normalizacaoParametros, dados)
p = list()
for (i in 1:15) {
  p[[i]] = ggplot(dados, aes_string(y=names(dados)[i])) + geom_boxplot(fill = i) +
    theme(legend.position="none")
```

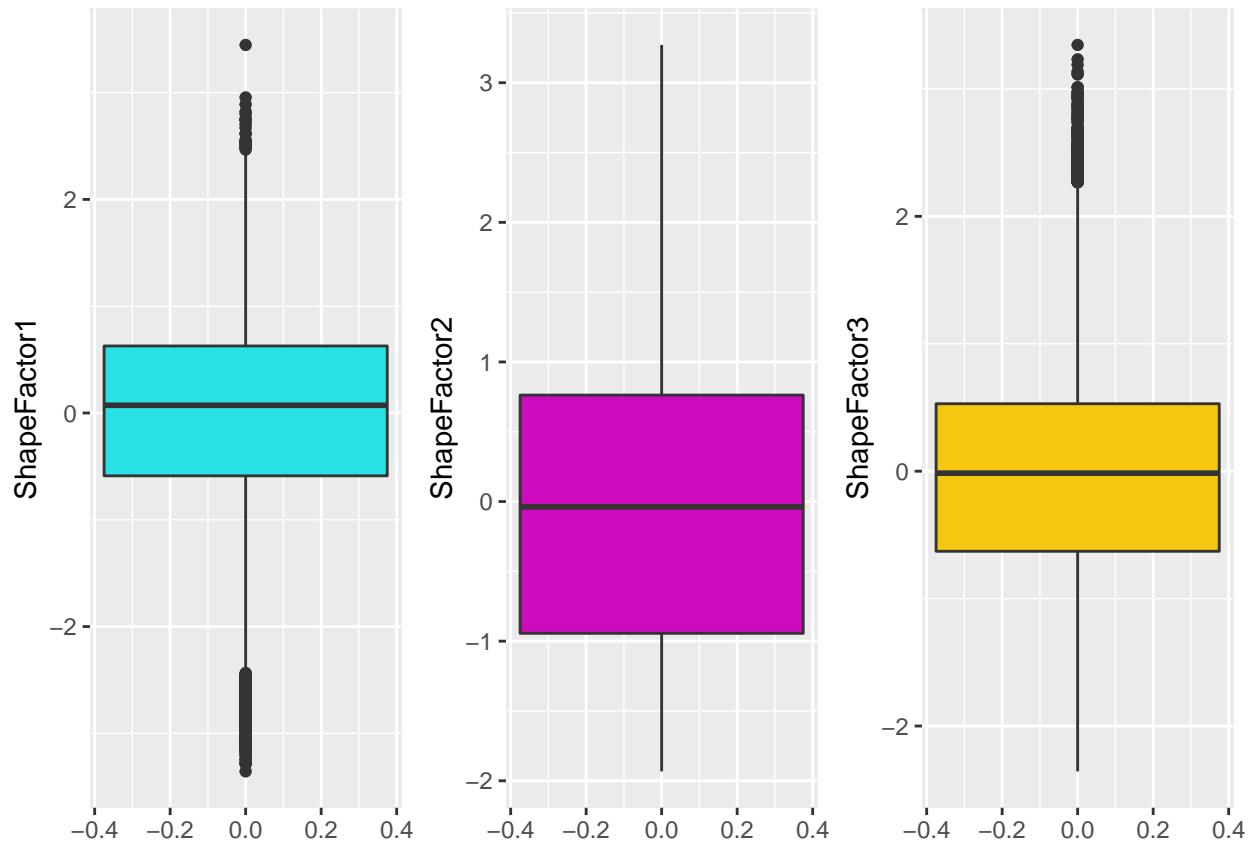
```
if((i==3) || (i==6) || (i==9) || (i==12) || (i==15)){
  do.call(grid.arrange,c(p[(i-2):i],ncol=3))
}
}
```











```
summary(dados)
```

```

##      Area      Perimeter MajorAxisLength MinorAxisLength
## Min. :-1.1127  Min. :-1.5425  Min. :-1.5933  Min. :-1.7736
## 1st Qu.:-0.5702 1st Qu.:-0.7082 1st Qu.:-0.7800 1st Qu.:-0.5876
## Median :-0.2863 Median :-0.2816 Median :-0.2714 Median :-0.2188
## Mean   : 0.0000 Mean   : 0.0000 Mean   : 0.0000 Mean   : 0.0000
## 3rd Qu.: 0.2825 3rd Qu.: 0.5690 3rd Qu.: 0.6576 3rd Qu.: 0.3282
## Max.   : 6.8738 Max.   : 5.2736 Max.   : 4.8862 Max.   : 5.7355
##
##      AspectRatio      Eccentricity ConvexArea      EquivDiameter
## Min. :-2.2636  Min. :-5.7819  Min. :-1.1111  Min. :-1.5516
## 1st Qu.:-0.6119 1st Qu.:-0.3801 1st Qu.:-0.5728 1st Qu.:-0.6421
## Median :-0.1302 Median : 0.1472 Median :-0.2885 Median :-0.2472
## Mean   : 0.0000 Mean   : 0.0000 Mean   : 0.0000 Mean   : 0.0000
## 3rd Qu.: 0.5021 3rd Qu.: 0.6475 3rd Qu.: 0.2863 3rd Qu.: 0.4458
## Max.   : 3.4339 Max.   : 1.7448 Max.   : 7.0359 Max.   : 5.3451
##
##      Extent      Solidity roundness      Compactness
## Min. :-3.9607  Min. :-14.5689  Min. :-6.4460  Min. :-2.5811
## 1st Qu.:-0.6336 1st Qu.:-0.3160 1st Qu.:-0.6920 1st Qu.:-0.6059
## Median : 0.2063 Median : 0.2446 Median : 0.1659 Median : 0.0229
## Mean   : 0.0000 Mean   : 0.0000 Mean   : 0.0000 Mean   : 0.0000
## 3rd Qu.: 0.7562 3rd Qu.: 0.6159 3rd Qu.: 0.7323 3rd Qu.: 0.5575
## Max.   : 2.3726 Max.   : 1.6167 Max.   : 1.9725 Max.   : 3.0373

```

```

##          ShapeFactor1      ShapeFactor2      ShapeFactor3      ShapeFactor4
##  Min.   :-3.35603   Min.   :-1.93292   Min.   :-2.35617   Min.   :-10.8500
##  1st Qu.:-0.58838   1st Qu.:-0.94387   1st Qu.:-0.62863   1st Qu.:-0.3116
##  Median : 0.07231   Median :-0.03762   Median :-0.01562   Median : 0.3029
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.62749   3rd Qu.: 0.76244   3rd Qu.: 0.52948   3rd Qu.: 0.6457
##  Max.   : 3.44642   Max.   : 3.27086   Max.   : 3.34535   Max.   : 1.0693
##
##          Class
##  BARBUNYA:1322
##  BOMBAY   : 522
##  CALI     :1630
##  DERMASON:3546
##  HOROZ    :1928
##  SEKER    :2027
##  SIRA     :2636

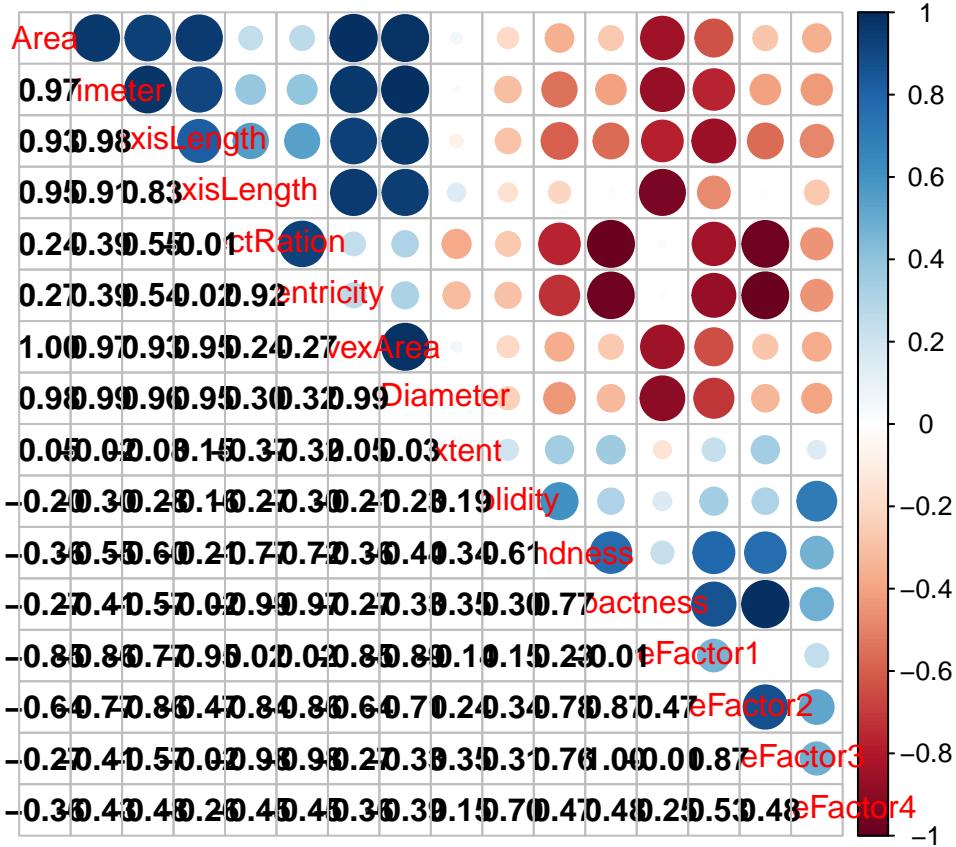
```

5 Realize a seleção de características (correlação). Plote o gráfico de correlação. Liste as características que foram removidas.

```

dados = dadosMain
rotulos = dados[,17]
dados = dados[,-17]
matrizCorrelacao = cor(dados)
indicesCorrelacaoForte = findCorrelation(matrizCorrelacao, cutoff=0.95)
corrplot.mixed(matrizCorrelacao,lower.col="black")

```



```

print(colnames(dados[,indicesCorrelacaoForte]))

## [1] "MajorAxisLength" "Perimeter"      "EquivDiameter"   "ConvexArea"
## [5] "ShapeFactor3"    "Compactness"    "MinorAxisLength"

if (length(indicesCorrelacaoForte) > 0)
  dados[,indicesCorrelacaoForte] = NULL

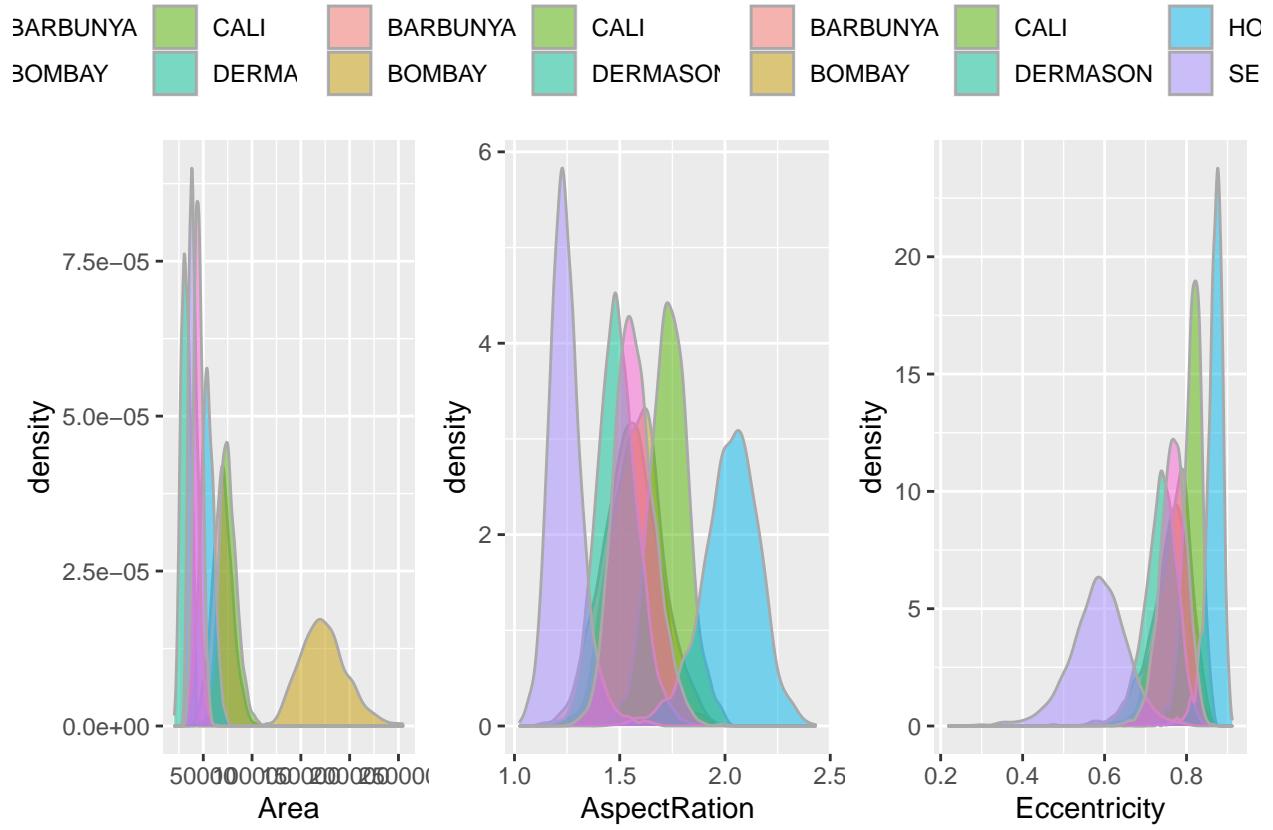
```

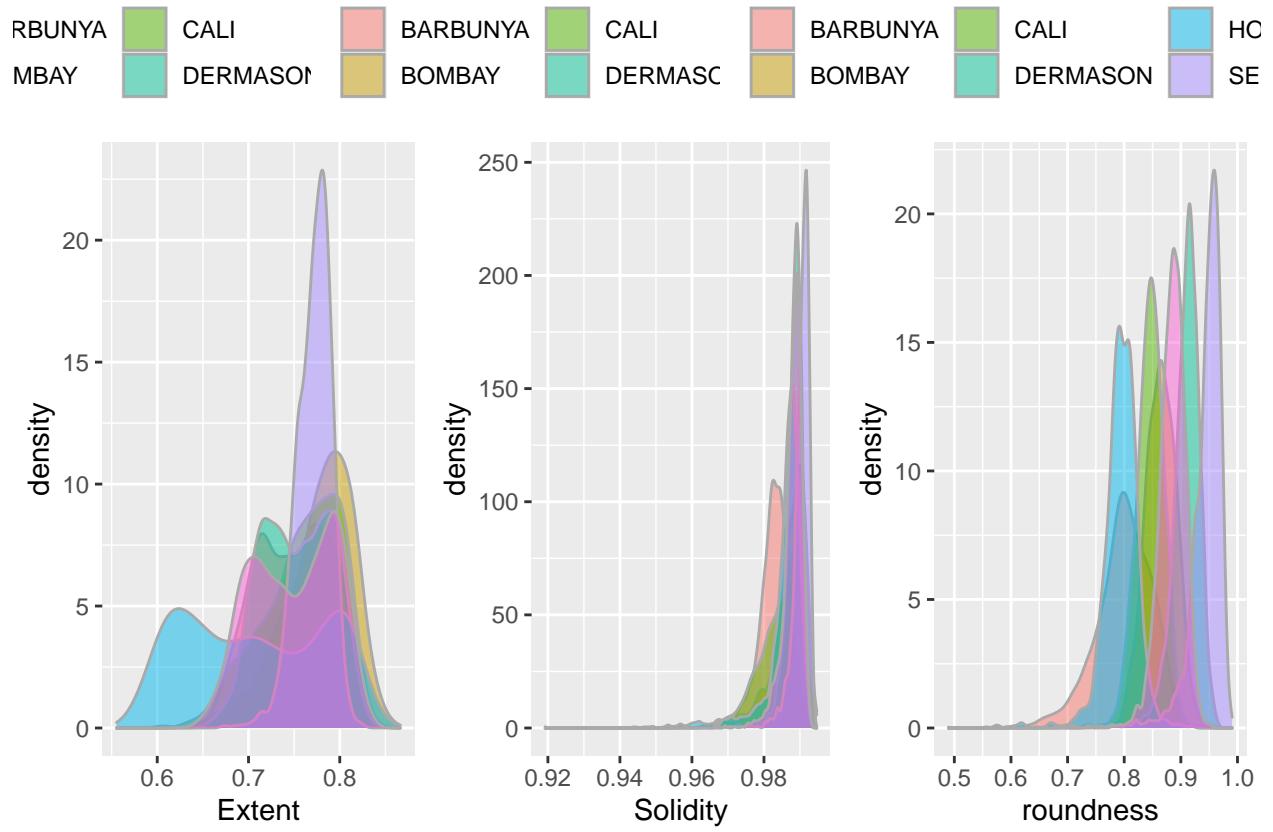
6 Plote um gráfico boxplot ou de densidade por variável x classe (organize em 3 colunas). Discuta qual é a variável que teria maior poder de discriminação? Existe alguma classe que pode ser classificada mais facilmente? Justifique a sua escolha.

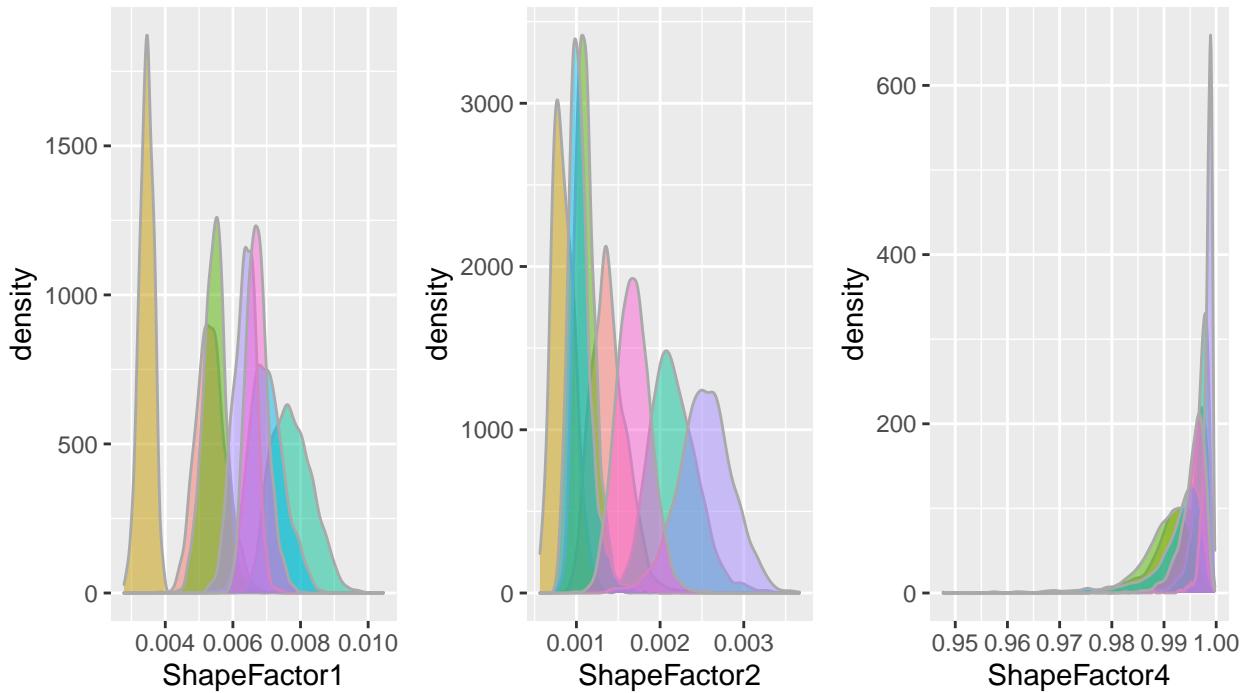
```

p = list()
dados$Class = dadosMain$Class
for(i in 1:length(dados)){
  p[[i]] = ggplot(dados, aes_string(x=names(dados)[i], fill="Class")) +
    geom_density(alpha=0.5, color="darkgray") +
    theme(legend.position="top", legend.title = element_blank())
  if((i==3) || (i==6) || (i==9) || (i==12) || (i==15)){
    do.call(grid.arrange, c(p[(i-2):i], ncol=3))
  }
}

```



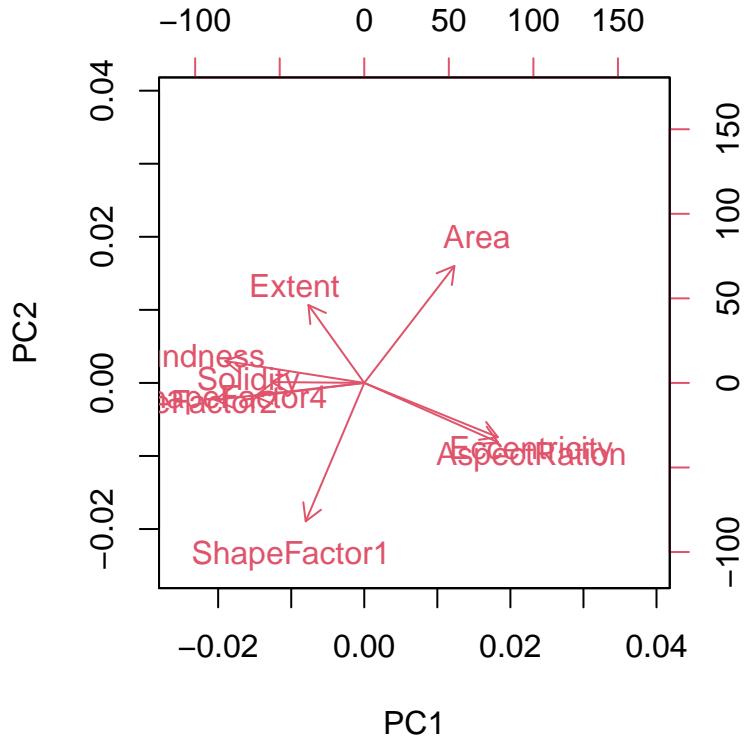




ShapeFactor2 é a variável que teria maior poder de discriminação, pois é a variável que possui maior dispersão de densidade entre as classes. Dentre as classes existentes, a Classe “BOMBAY” é a que pode mais facilmente ser classificada pois tanto nas variáveis “Area” e “ShapeFactor1”, esta classe tem grande discrepância das demais classes.

7 Realize a projeção do dataset utilizando PCA. Explique as características dos componentes principais estimados. O que se pode explicar sobre os componentes principais utilizando o gráfico biplot. Apresente as características básicas (summary) dos dados.

```
dados = dados[,-length(dados)]
pca = prcomp(dados, center=TRUE, scale=TRUE)
biplot(pca, xlab = rep("", nrow(dados)))
```



```
summary(pca)
```

```
## Importance of components:
##              PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation   2.1249 1.3633 1.0695 0.87440 0.65139 0.37872 0.31486
## Proportion of Variance 0.5017 0.2065 0.1271 0.08495 0.04715 0.01594 0.01102
## Cumulative Proportion 0.5017 0.7082 0.8353 0.92022 0.96736 0.98330 0.99431
##                  PC8     PC9
## Standard deviation   0.21469 0.07134
## Proportion of Variance 0.00512 0.00057
## Cumulative Proportion 0.99943 1.00000
```

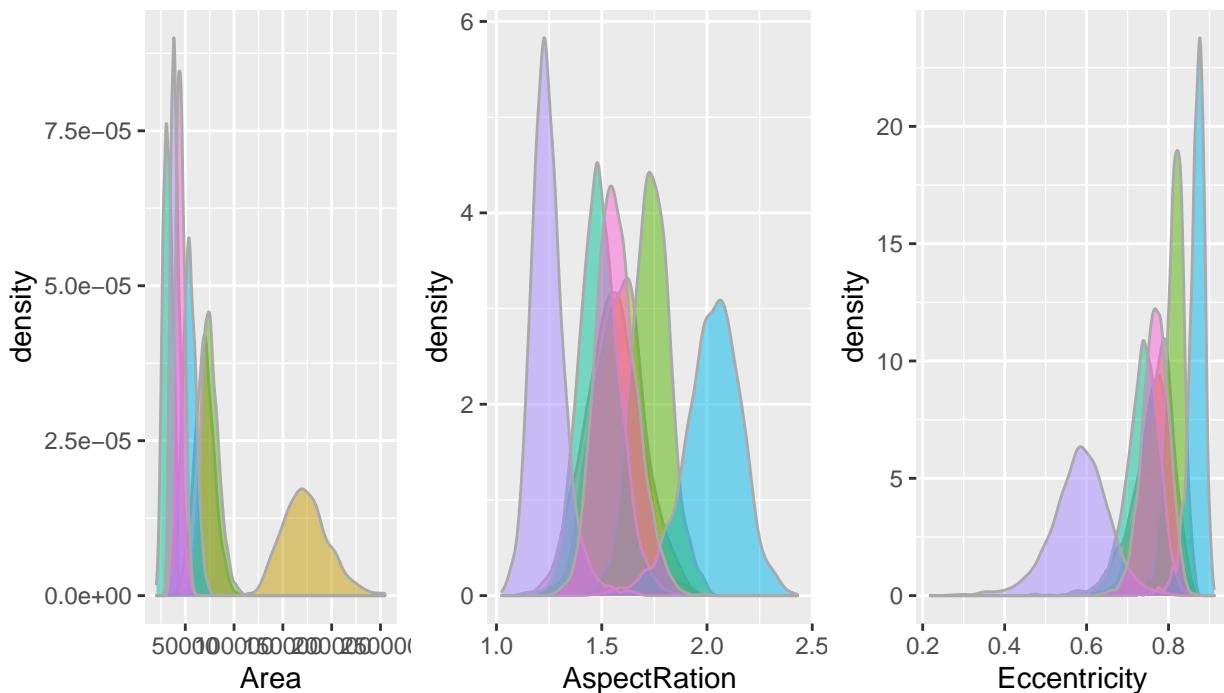
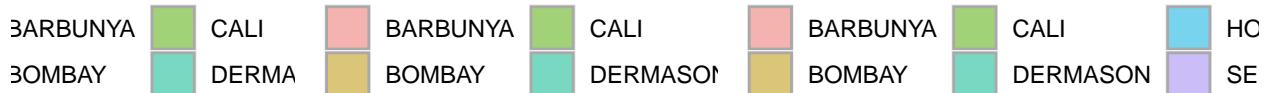
```
colnames(dados)
```

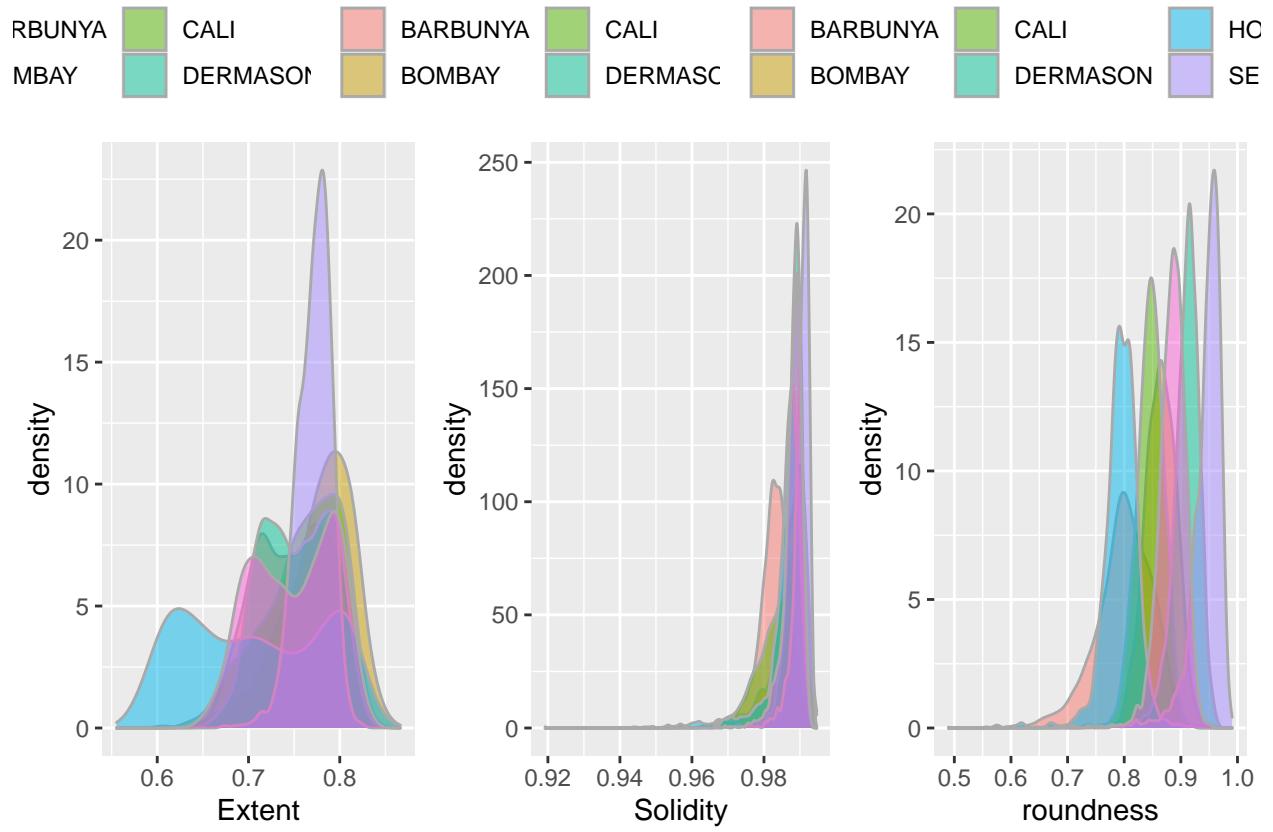
```
## [1] "Area"          "AspectRatio"    "Eccentricity"  "Extent"        "Solidity"
## [6] "roundness"     "ShapeFactor1"  "ShapeFactor2"  "ShapeFactor4"
```

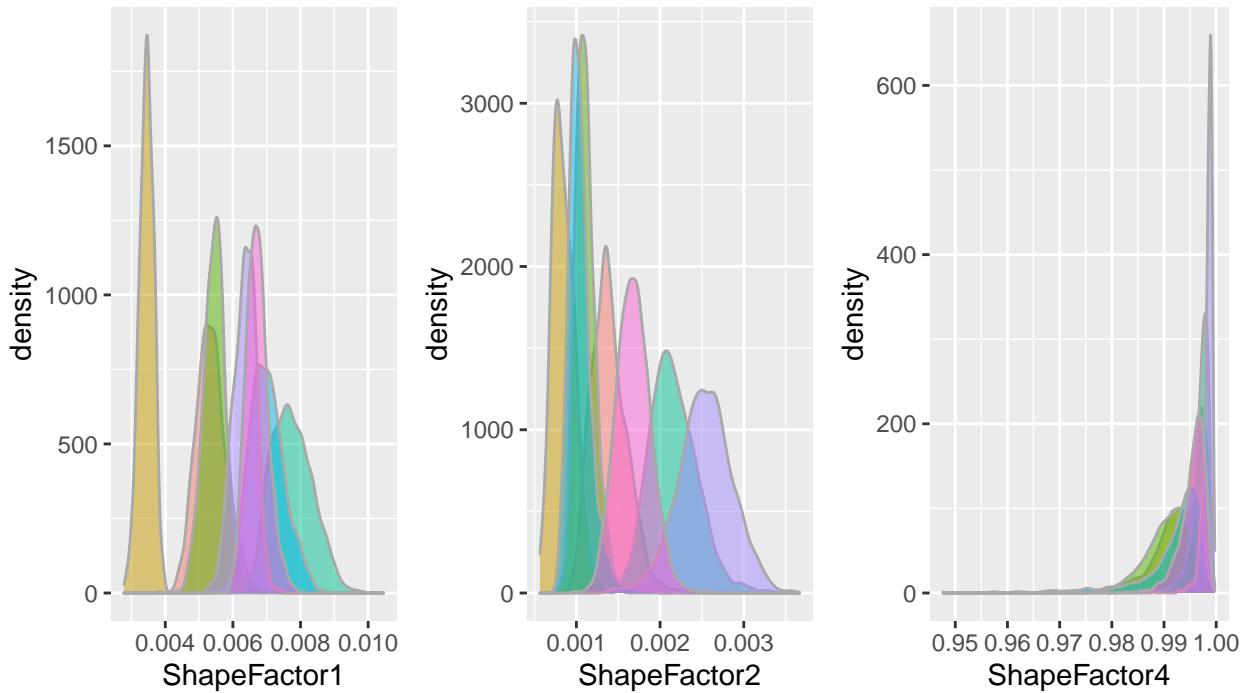
As variáveis “AspectRation”, “Eccentricity”, “Solidity” e “roundness” são as que influenciam mais no componente principal 1. Podem-se dizer que as maiores medidas permitem discriminar melhor as classes. As variáveis “Area”, “ShapeFactor1” e “Extent” são as que influencia mais no componente principal 2. As variáveis “AspectRation” e “Eccentricity” são altamente correlacionadas, pois o ângulo entre elas é muito pequeno. O mesmo acontece para “Solidity” e “roundness”. As variáveis “Area”, “ShapeFactor1” e “Extent” não são correlacionadas com as demais, nem com elas mesmas, porque apresentam um ângulo muito abertos.

8 Analise o dataset projetado com o auxílio do gráfico de boxplot por classe (igual ao do item 6). Compare com o resultado do item 6. Se quiser, pode gerar um gráfico de espalhamento para auxiliar na explicação.

```
p = list()
dados$Class = dadosMain$Class
for(i in 1:length(dados)){
  p[[i]] = ggplot(dados, aes_string(x=names(dados)[i], fill="Class")) +
    geom_density(alpha=0.5, color="darkgray") +
    theme(legend.position="top", legend.title = element_blank())
  if((i==3) || (i==6) || (i==9) || (i==12) || (i==15)){
    do.call(grid.arrange,c(p[(i-2):i], ncol=3))
  }
}
```

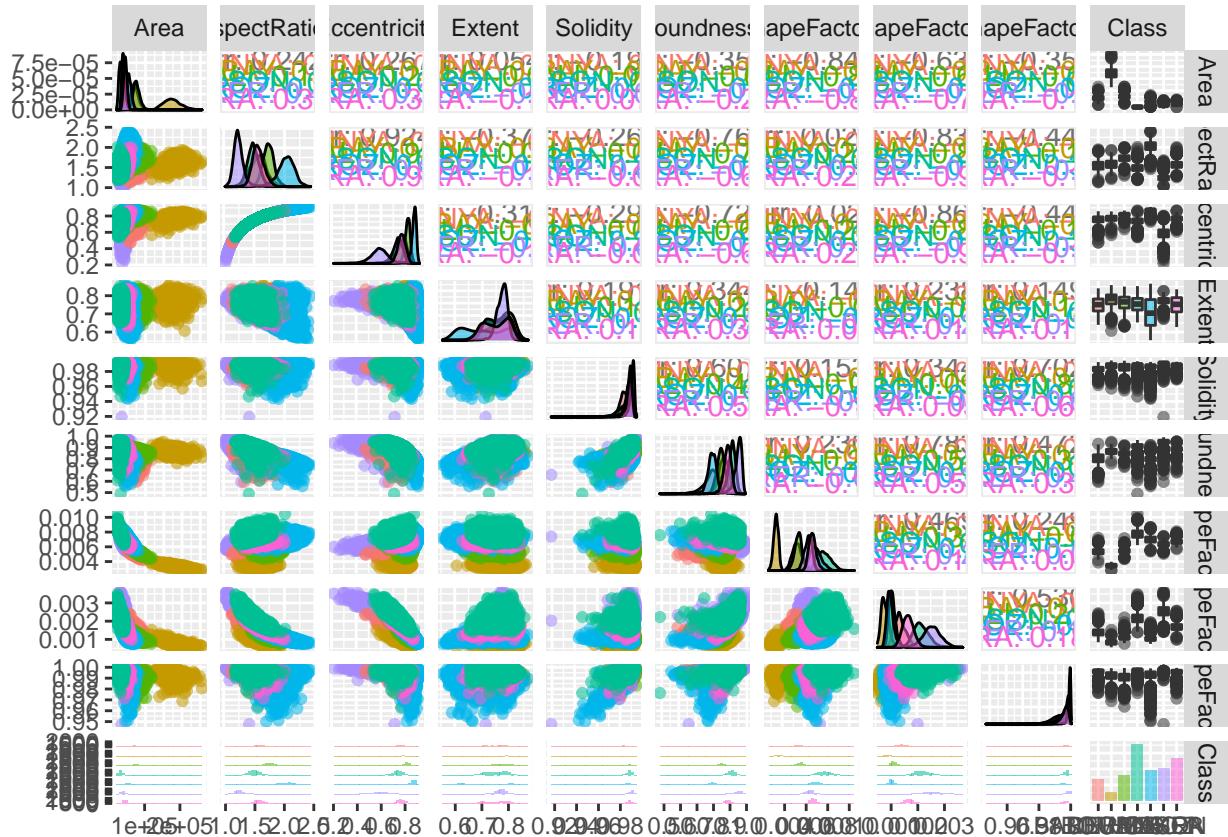






```
ggpairs(dados,aes(colour=rotulos, alpha=0.1))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



9 É possível reduzir a dimensionalidade dos dados? Explique como!

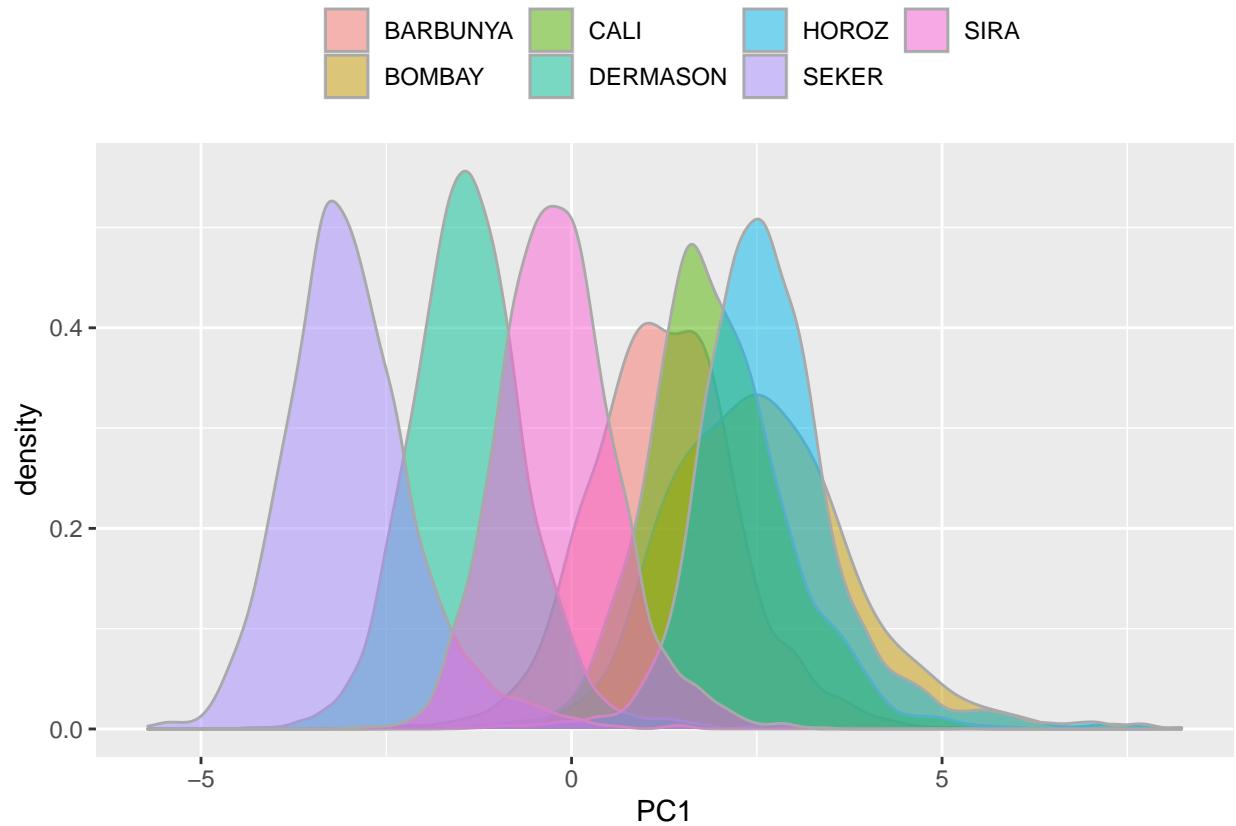
```
numeroComponentes = min(which(summary(pca)$importance[3,] > 0.95))
dados = predict(pca,dados)[,1:numeroComponentes]
dados = data.frame(dados)
```

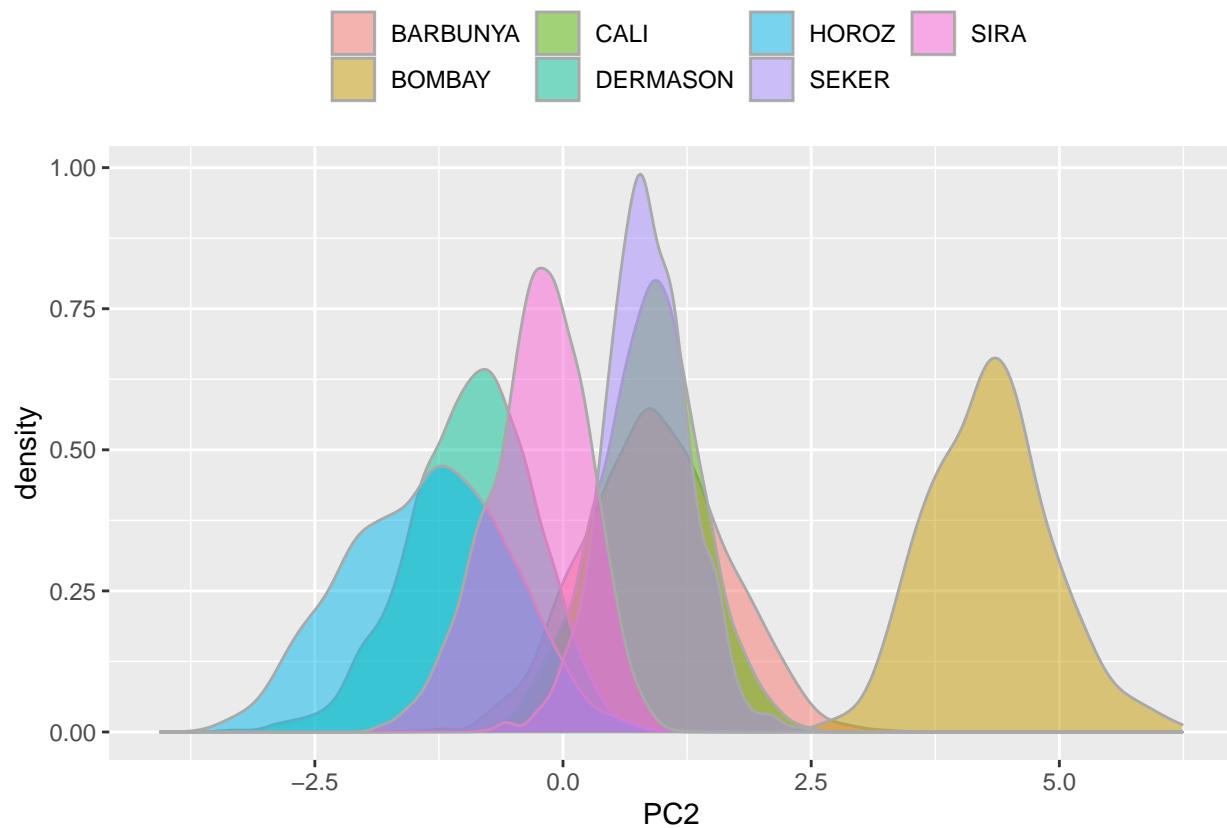
Uma maneira de fazer a redução da dimensionalidade é realizar a Seleção dos autovetores que explicam pelo menos 95% da variância dos dados.

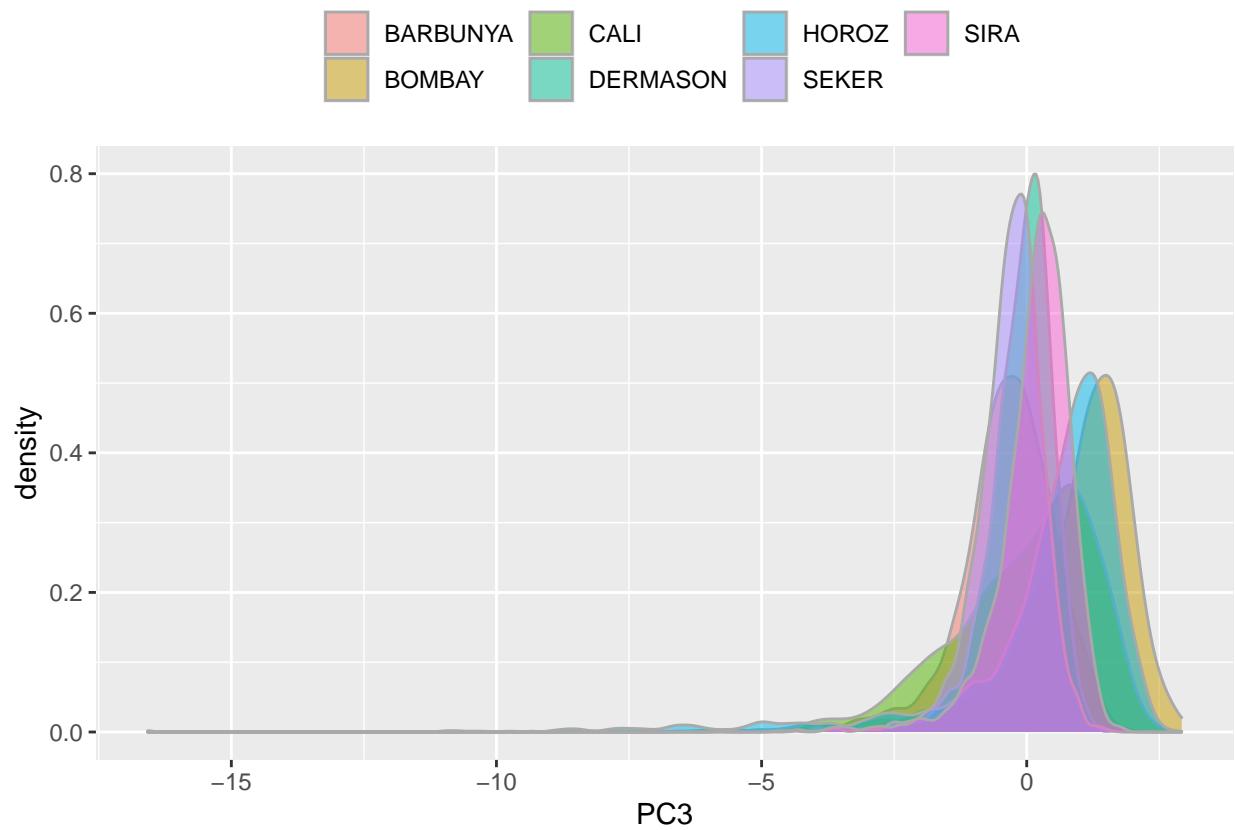
10 Analise o dataset reduzido com o auxílio do gráfico de boxplot por classe (igual ao do item 6). Compare com o resultado do item 6 e do item 8. Se quiser, pode gerar um gráfico de espalhamento para auxiliar na explicação.

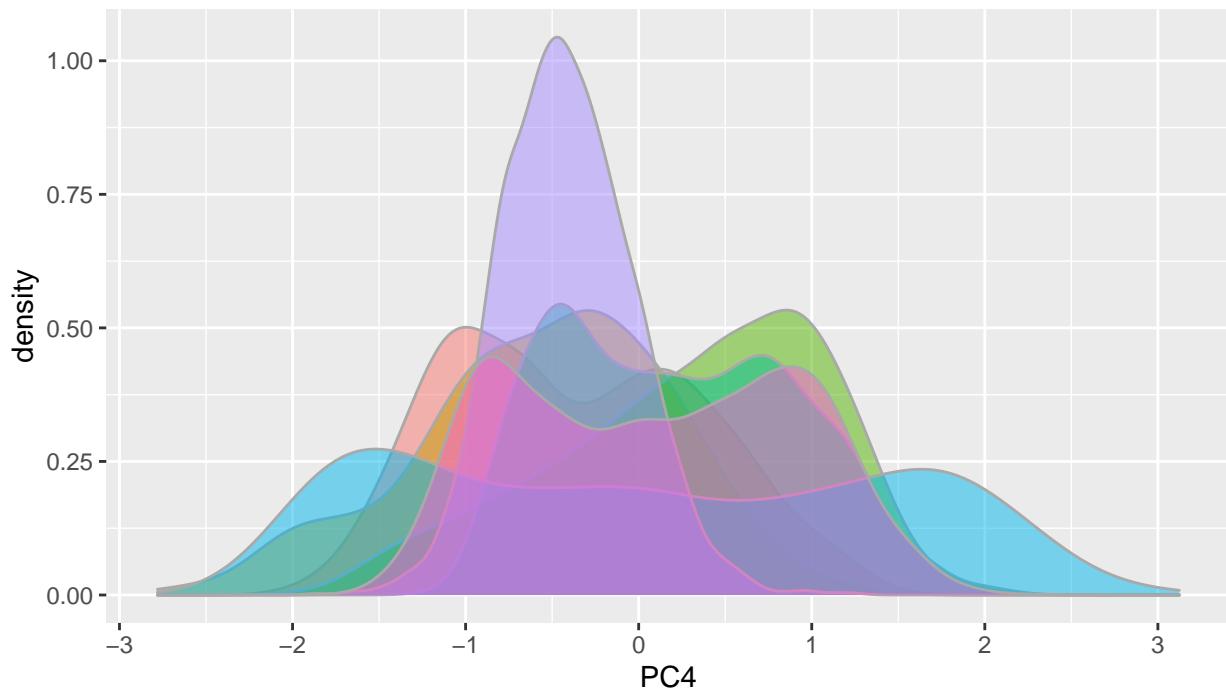
```
p = list()
dados$Class = dadosMain$Class
for(i in 1:length(dados)){
  p[[i]] = ggplot(dados, aes_string(x=names(dados)[i], fill="Class")) +
    geom_density(alpha=0.5, color="darkgray") +
    theme(legend.position="top", legend.title = element_blank())
  if(i<6){
    do.call(grid.arrange,c(p[i],ncol=1))
```

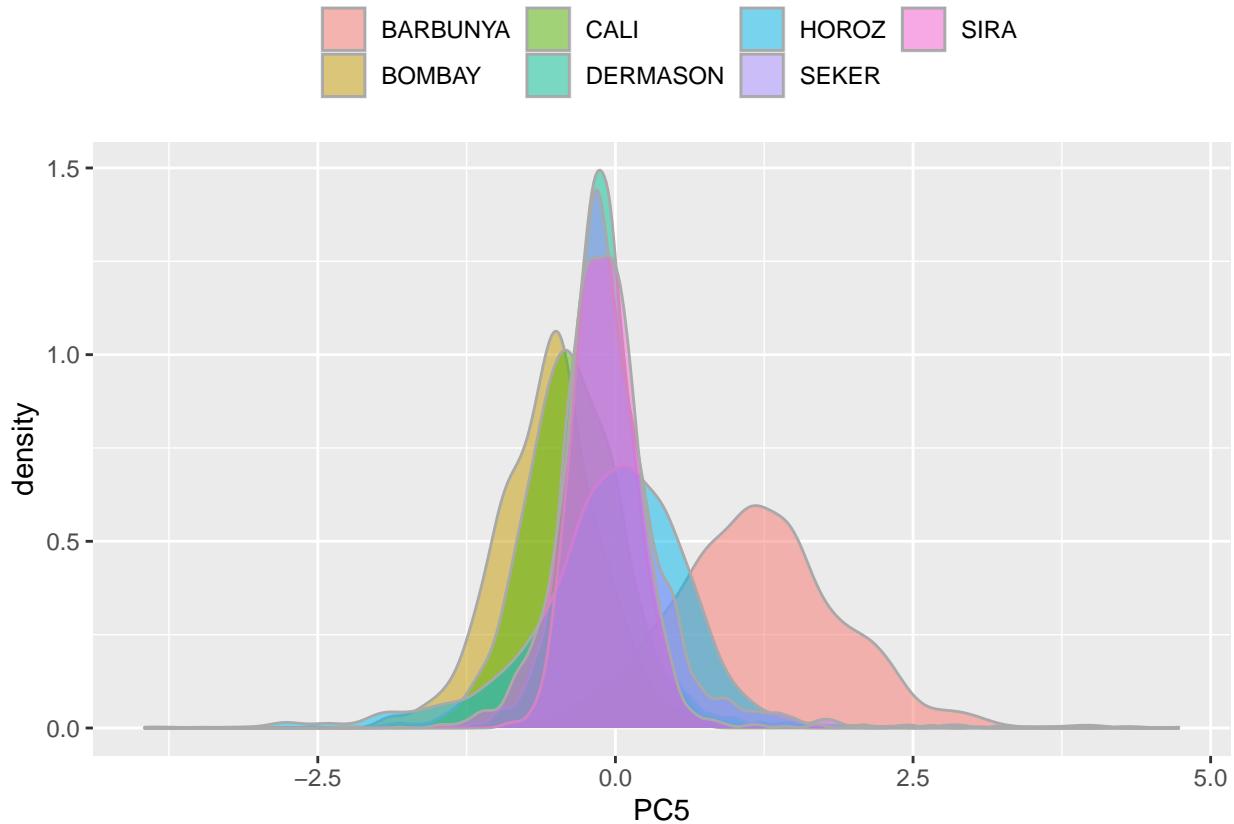
```
    }  
}
```





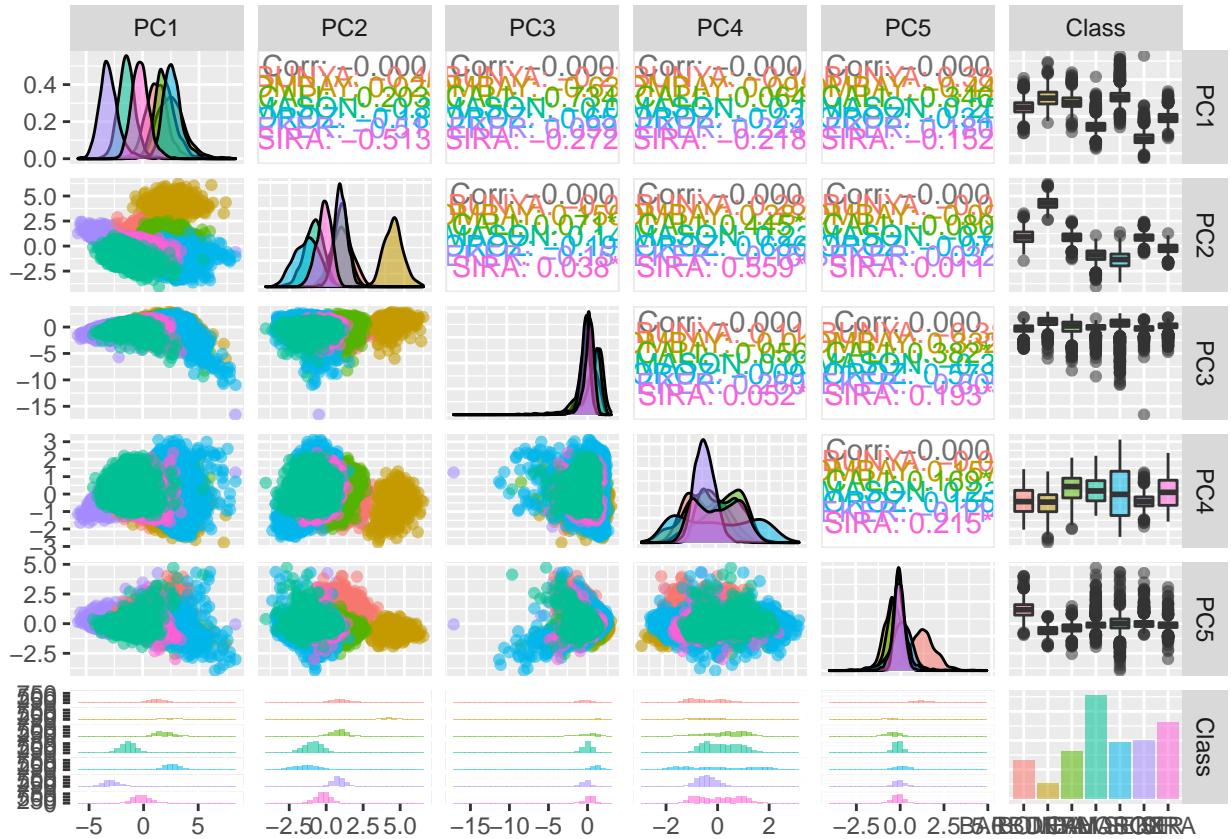






```
ggpairs(dados,aes(colour=rotulos, alpha=0.1))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



11 Após ter analisado estas informações, quais considerações você faz sobre este conjunto de dados (ou tarefa)?

Com relação à classificação das sementes de feijão, as características de dimensão e forma das variedades de feijão não possuem características discriminatórias externas, o que torna esse processo de classificação complexo. Mas utilizando diversas maneiras de realizar reduções sobre a dimensionalidade dos dados, assim permitindo uma melhor maneira de classificar os dados.