Exercices avec le logiciel R

Épreuve M1 A9

Pr Jean R. Lobry

Contrôle - 15 janvier 2008 - Durée 1h30

M1 - A9 - 15 janvier 2008 *Tous documents autorisés - échanges strictement interdits*

1 Répondre directement sur la feuille

Nom :
Prénom :
Numéro carte étudiant :

2 Le diabète des indiens Pima

2.1 Introduction

Les données [2] concernent 768 femmes adultes (âgées d'au moins 21 ans) de la tribu indienne Pima (Akimel O'odham) vivant près de Phoenix, Arizona, USA. La description des données [1] est la suivante :

1. `pregnant` - Nombre de grossesses.
2. `glucose` - Glycémie après un test de tolérance au glucose (cg/l).
3. `diastolic` - Tension artérielle diastolique (mm Hg).
4. `triceps` - Indice d'obésité (mm).
5. `insulin` - Concentration en insuline (mu U/ml).
6. `bmi` - Indice de Masse corporelle (poids en kg/(taille en m)^2)
7. `diabetes` - Indice d'antécédents familiaux pour le diabète.
8. `age` - Age (années)
9. `test` - Variable indicatrice de l'absence (0) ou présence (1) de diabète selon les critères de l'OMS.

2.2 Importation des données

```
pima1 <- read.table("http://pbil.univ-lyon1.fr/R/donnees/pima.txt", sep = "\t", header = TRUE)
names(pima1)
```

```
[1] "pregnant"  "glucose"   "diastolic" "triceps"   "insulin"   "bmi"
[7] "diabetes"  "age"       "test"
```

1. Reprise de `http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes`, voir [1].

À quoi sert la fonction `read.table()` et les arguments `sep` et `header` ?

Réponse :

2.3 Analyse univariée

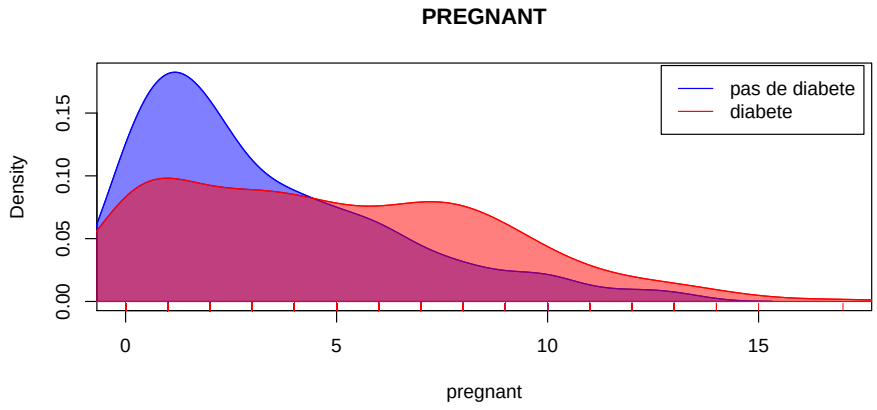
On définit la fonction utilitaire suivante pour représenter les données :

```
mafig <- function(quoi, posleg = "topleft", data = pima1){
        x0 <- data[data[,"test"] == 0, quoi]
        x1 <- data[data[,"test"] == 1, quoi]
  dst0 <- density(x0, na.rm = TRUE)
          dst1 <- density(x1, na.rm = TRUE)
          plot(dst0, col = "blue", main = toupper(quoi),
            xlim = range(data[,quoi], na.rm = TRUE), xlab = quoi,
            ylim = c(0, max(dst0$y, dst1$y, na.rm = TRUE)))
          lines(dst1, col = "red")
          polycurve <- function(x, y, base.y = min(y), ...) {
    polygon(x = c(min(x), x, max(x)), y = c(base.y, y, base.y),
...)
  }
  polycurve(dst0$x, dst0$y, base.y=0, col = rgb(0,0,1,0.5), border = "blue")
    polycurve(dst1$x, dst1$y, base.y=0, col = rgb(1,0,0,0.5), border = "red")

          legend(posleg, inset = 0.01,
          legend = c("pas de diabete", "diabete"),
          lty = 1, col = c("blue","red"))
          rug(x0, col = "blue")
          rug(x1, col = "red")
}
```

2.3.1 Nombre de grossesses

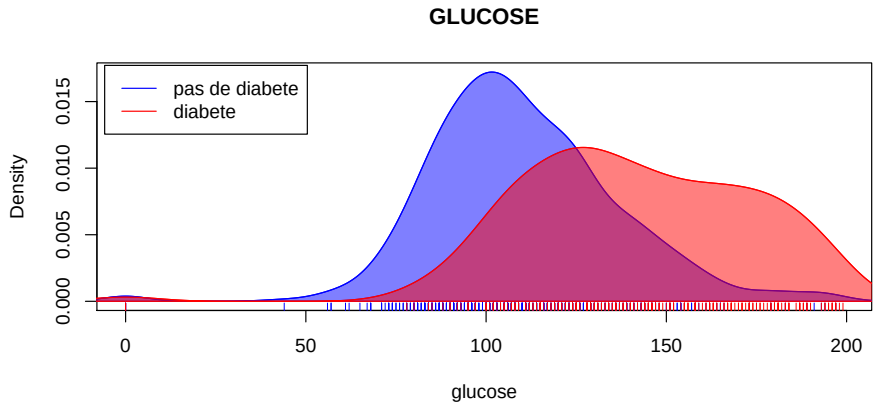
```
mafig("pregnant", "topright")
```



Commentaires :

2.3.2 Glycémie

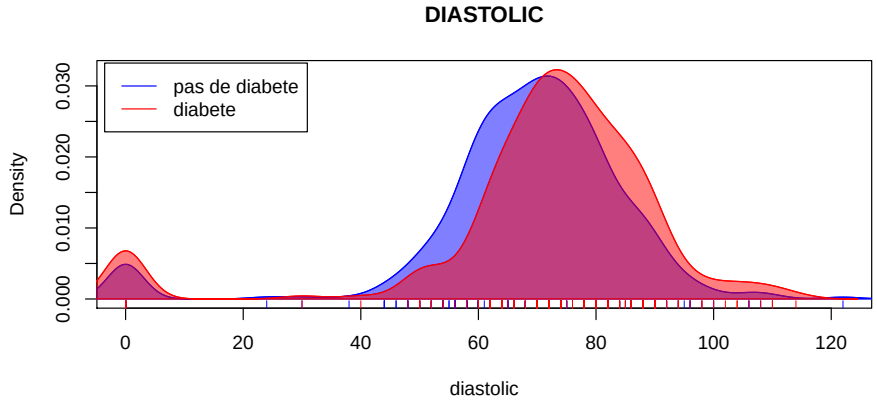
```
mafig("glucose")
```



Commentaires :

2.3.3 Tension

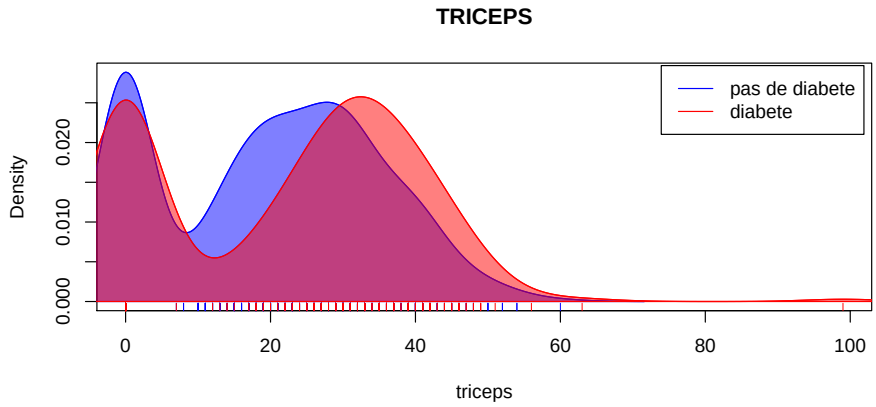
```
mafig("diastolic")
```



Commentaires :

2.3.4 Obésité

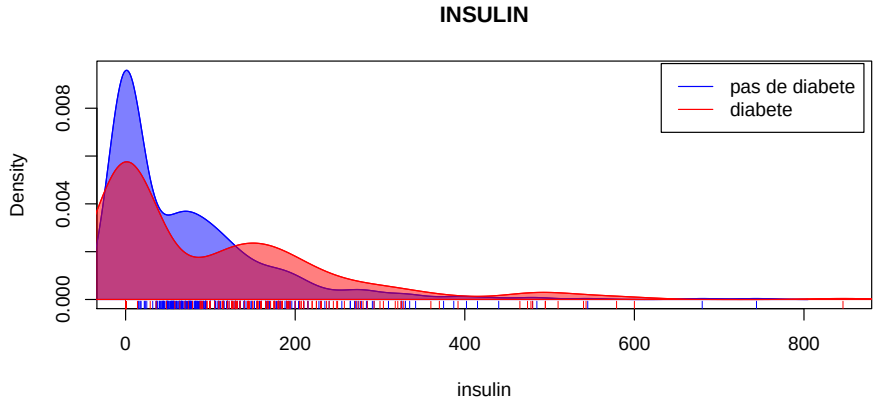
```
mafig("triceps", "topright")
```



Commentaires :

2.3.5 Insuline

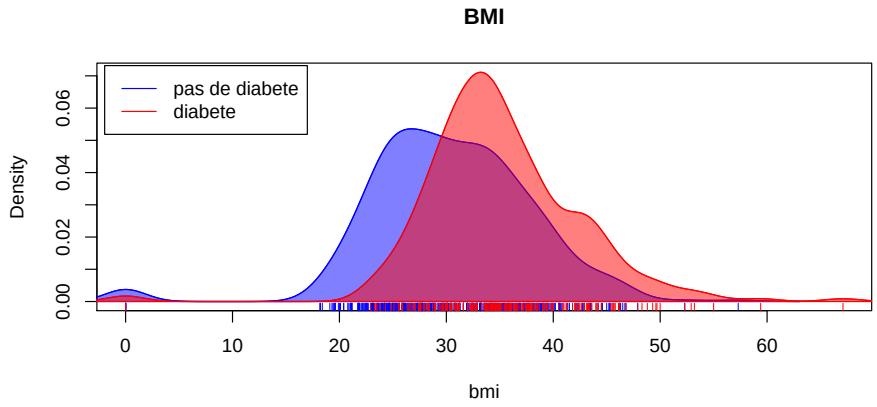
```
mafig("insulin", "topright")
```



Commentaires :

2.3.6 Indice de masse corporelle

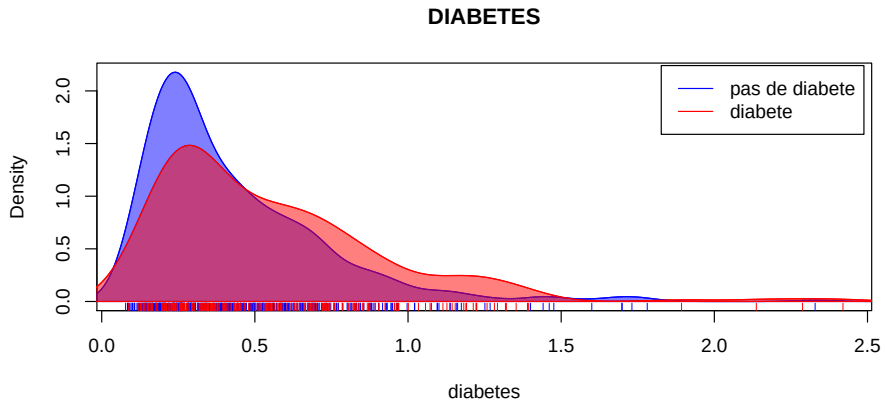
```
mafig("bmi")
```



Commentaires :

2.3.7 Antécédents familliaux

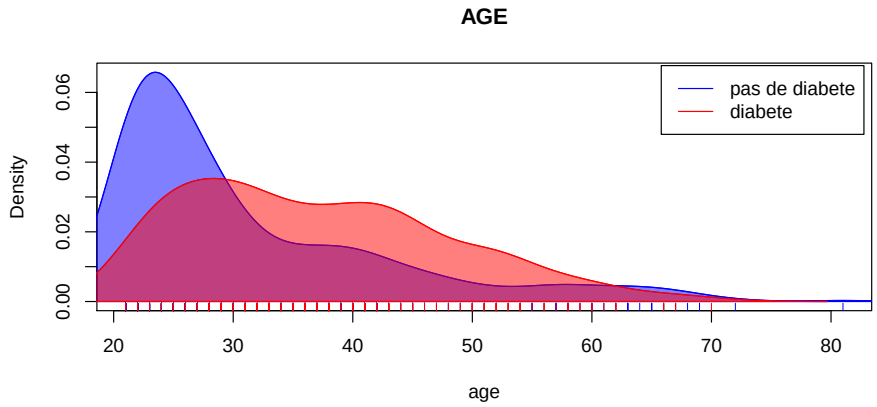
```
mafig("diabetes", "topright")
```



Commentaires :

2.3.8 Age

```
mafig("age", "topright")
```



Commentaires :

2.4 Pré-traitement des données

On construit le jeu de données `pima2` de la façon suivante :

```
pima2 <- pima1
for(j in c("glucose", "diastolic", "triceps", "insulin",
"bmi", "diabetes")){
  pima2[pima2[,j] == 0, j] <- NA
}
```

La représentation graphique de `pima2` est donnée dans la figure 1, elle a été obtenue avec le code R suivant :

```
par(mfrow = c(4,2))
for(i in names(pima2)[1:8]) mafig(i, "topright", data = pima2)
```

Expliquez pourquoi on a construit le jeu de données `pima2`.

Réponse :

2.5 Analyse multivariée

2.5.1 Première analyse multivariée

```
pima3 <- pima2[complete.cases(pima2),]
dim(pima3)
```

```
[1] 392   9
```

À quoi correspond le jeu de données `pima3` ?

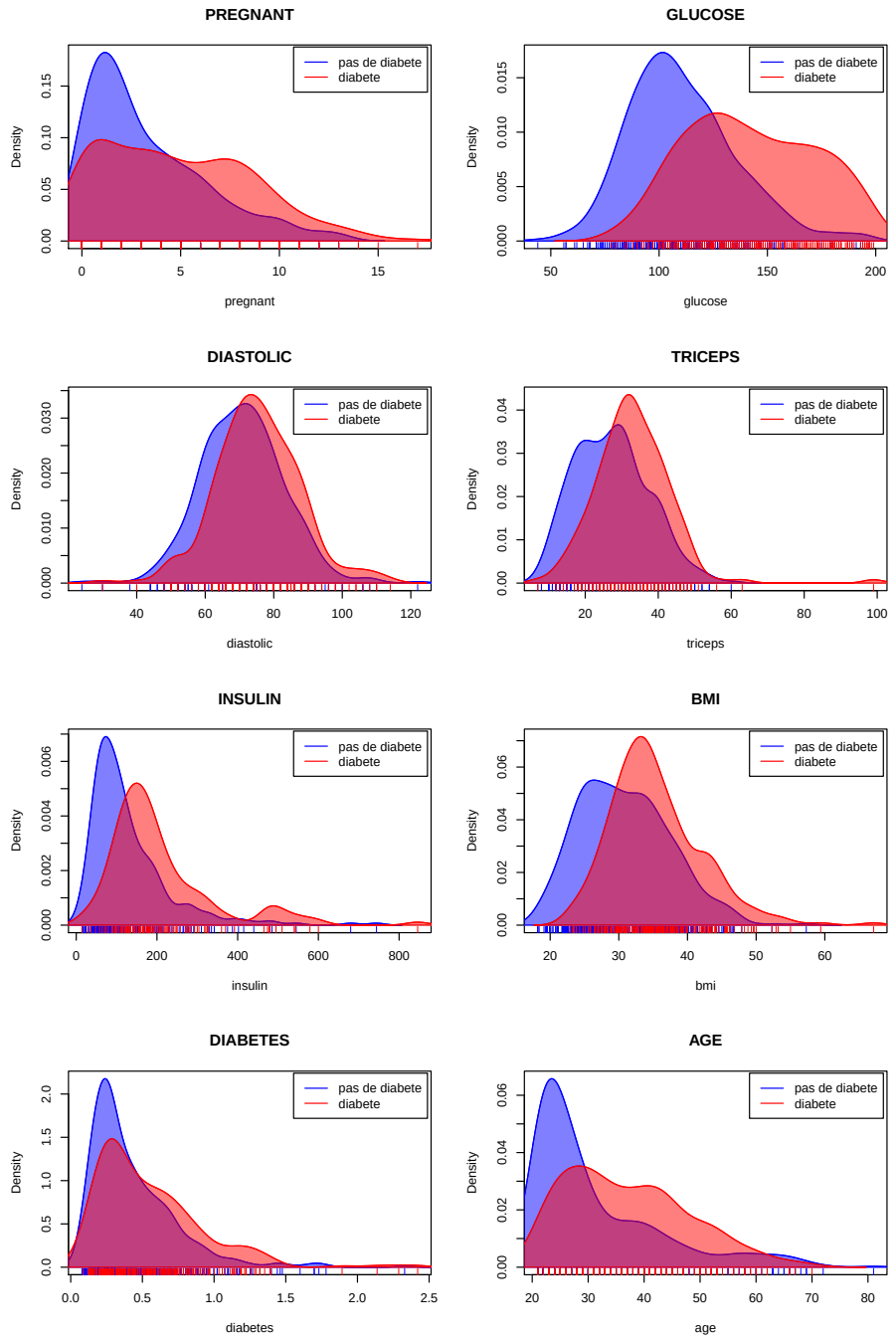
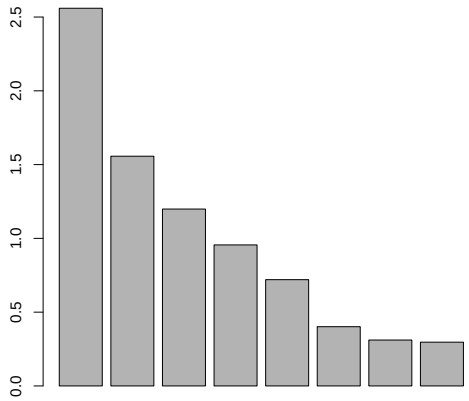


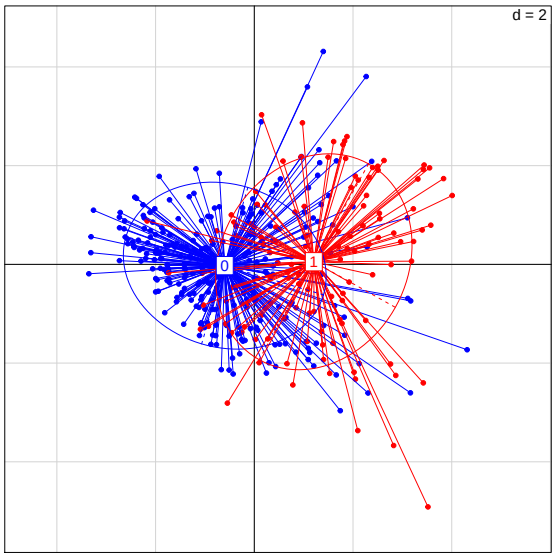
FIGURE 1 – Représentation univariée du jeu de données `pima2`.

Réponse :

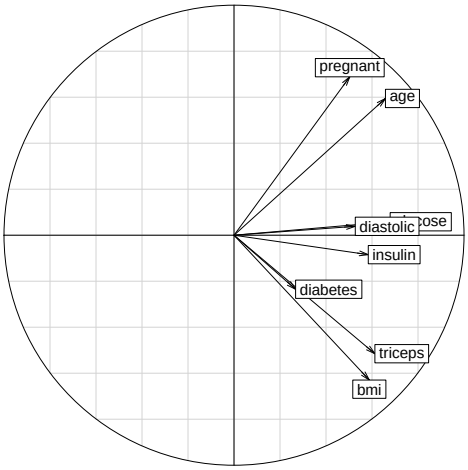
```
library(ade4)
acp1 <- dudi.pca(pima3[, 1:8], scann = FALSE, nf = 2)
barplot(acp1$eig, col = grey(0.7))
```




```
s.class(acp1$li, as.factor(pima3$test), col = c("blue", "red"))
```




```
s.corcircle(acp1$co)
```



Interpréter le premier plan factoriel de l'ACP pour le jeu de données `pima3`.

Réponse :

2.5.2 Deuxième analyse multivariée

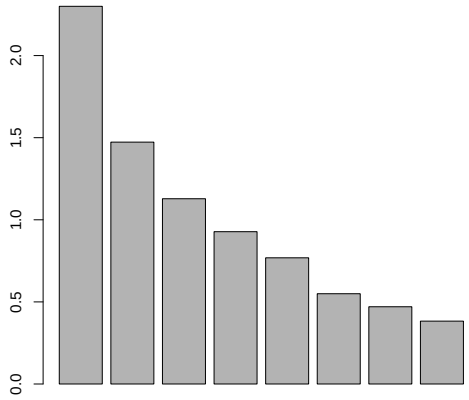
```
pima4 <- apply(pima2, 2, function(x) ifelse(is.na(x),
mean(x, na.rm = TRUE), x))
pima4 <- as.data.frame(pima4)
dim(pima4)
```

```
[1] 768   9
```

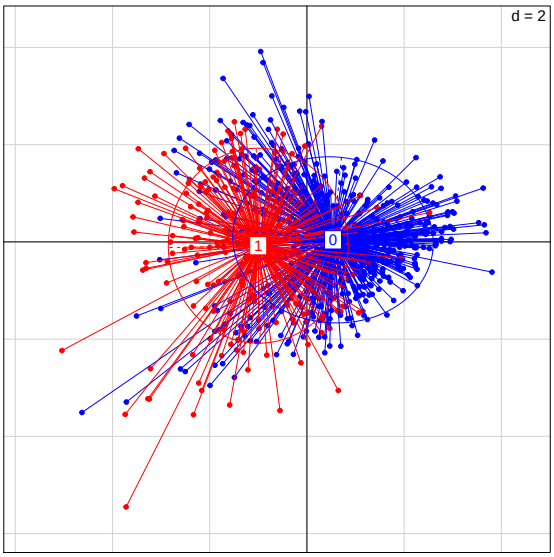
À quoi correspond le jeu de données `pima4` ?

Réponse :

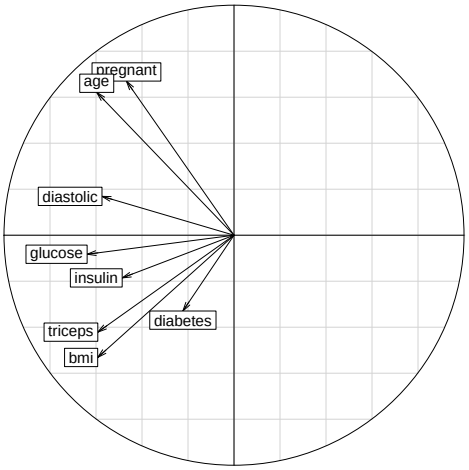
```
acp2 <- dudi.pca(pima4[, 1:8], scann = FALSE, nf = 2)
barplot(acp2$eig, col = grey(0.7))
```




```
s.class(acp2$li, as.factor(pima4$test), col = c("blue", "red"))
```




```
s.corcircle(acp2$co)
```



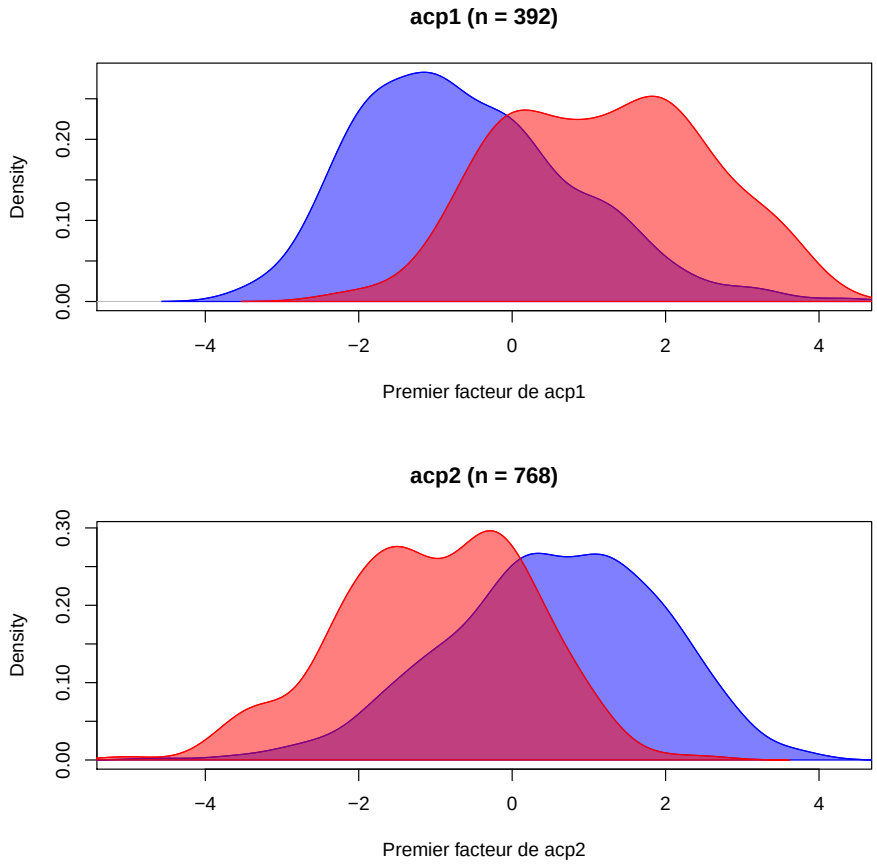
Interpréter le premier plan factoriel de l'ACP pour le jeu de données `pima4`.

Réponse :

2.5.3 Comparaisons des deux analyses multivariées

```
crange <- range(c(acp1$li[,1], acp2$li[,1]))
onefig <- function(acp = acp1, data = pima3, posleg = "topright"){
        x0 <- acp$li[,1][data[,"test"]==0]
        x1 <- acp$li[,1][data[,"test"]==1]
  dst0 <- density(x0, na.rm = TRUE)
          dst1 <- density(x1, na.rm = TRUE)
          plot(dst0, col = "blue", main = paste(substitute(acp),
          " (n = ", nrow(acp$li), ")", sep = ""),
            xlim = crange,
            xlab = paste("Premier facteur de", substitute(acp)),
            ylim = c(0, max(dst0$y, dst1$y, na.rm = TRUE)))
          lines(dst1, col = "red")
          polycurve <- function(x, y, base.y = min(y), ...) {
    polygon(x = c(min(x), x, max(x)), y = c(base.y, y, base.y),
...)
  }
  polycurve(dst0$x, dst0$y, base.y=0, col = rgb(0,0,1,0.5), border = "blue")
    polycurve(dst1$x, dst1$y, base.y=0, col = rgb(1,0,0,0.5), border = "red")

}
par(mfrow=c(2,1))
onefig()
onefig(acp2,pima4)
```



Que vous suggère cette représentation graphique ?

Réponse :

Références

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[2] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In R.A. Greenes, editor, *Symposium on Computer Applications and*

Medical Care, pages 261–265, Los Alamitos, CA, USA, 1988. IEEE Computer Society Press.