Contrôle continu, Master 1 MIASHS, 19 novembre 2015, 1h 25 minutes

Prénom, nom : correction

À faire :

- dans le nom de ce fichier, vous changez Prenom et Nom par votre prénom et votre nom ;
- vous répondez dans ce fichier ; vos réponses s'affichent en noir.
- à l'issue du contrôle, vous envoyez le **pdf** de ce fichier à philippe.preux@univ-lille3.fr

---

Exercice 1

Chargez le fichier se trouvant à l'url
http://www.grappa.univ-lille3.fr/~ppreux/ensg/miashs/m1/tps/cc/seeds.csv
Le dernier attribut (la colonne portant le numéro le plus élevé) correspond à la classe de la donnée.
Tous les autres attributs sont quantitatifs.

Question 1. comment faites-vous pour charger ce fichier dans R ?

```
seeds <- read.csv ("http://www.grappa.univ-lille3.fr/~ppreux/ensg/miashs/m1/tps/cc/seeds.csv")
```

Question 2. quelle commande R utilisez-vous pour connaître :
le nombre de lignes ?

```
nrow (seeds)
```

le nombre de colonnes (attributs) ?

```
ncol (seeds)
```

le nom des attributs ?

```
names (seeds)
```

Question 3.
Combien y a-t-il de classes ?
On peut faire :

```
table (seeds$variety)
```

il y a 3 classes

Quel est l'effectif de chacune des classes ?
70 données par classe.

Question 4. Comment déterminez-vous les données dont l'attribut area est supérieur à 20 et l'attribut perimeter est supérieur à 17 ?

```
which ((seeds$area > 20) & (seeds$perimeter > 17))
```

Comment faites-vous pour savoir combien il y en ?

```
length (which ((seeds$area > 20) & (seeds$perimeter > 17)))
```

Ces données sont-elles de la même classe ?
On regarde la classe des données :

```
seeds$variety [which ((seeds$area > 20) & (seeds$perimeter > 17))]
```

On conclut que oui, elles sont toutes de la classe 2.

Question 5. On veut réaliser un graphique de l'attribut 6 en fonction de l'attribut 2 en mettant chaque point en une couleur correspondant à la classe de la donnée, sans oublier les légendes.
Quelle(s) commande(s) tapez-vous pour créer ce graphique ?

L'attribut 6 en fonction de l'attribut 2 signifie que l'attribut 2 est en abscisses et l'attribut 6 en ordonnées.

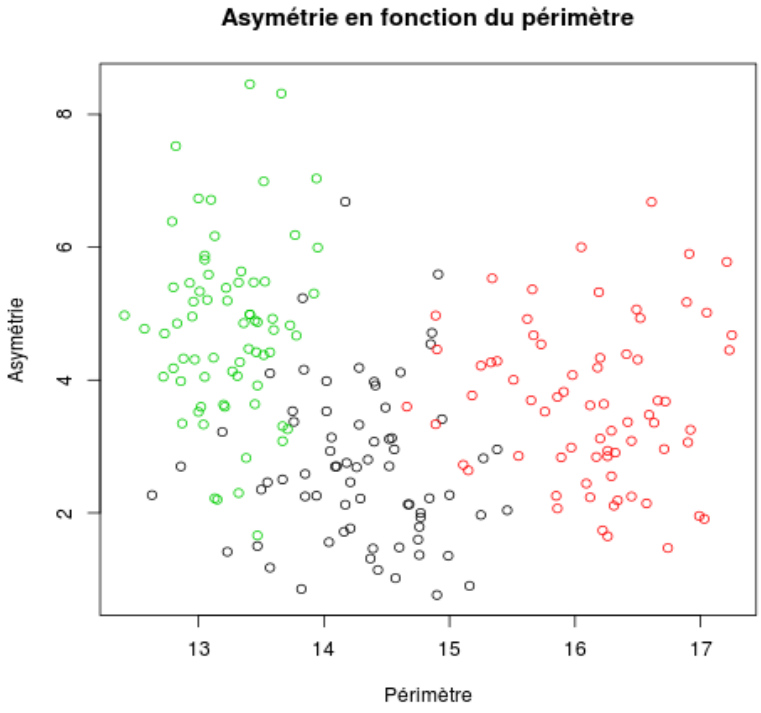
```
plot (seeds [, 2], seeds [, 6], col = seeds$variety,
      main = "Asymétrie en fonction du périmètre", xlab = "Périmètre",
      ylab = "Asymétrie")
```

Comment faites-vous pour que le graphique soit mis dans un fichier au format png ?

On fait :

```
png ("seeds.png")
plot (seeds [, 2], seeds [, 6], col = seeds$variety,
      main = "Asymétrie en fonction du périmètre", xlab = "Périmètre",
      ylab = "Asymétrie")
dev.off ()
```

Insérez-le ci-dessous.



Question 6. On veut réaliser une segmentation de ce jeu de données en 3 groupes.

Quel algorithme allez-vous utilisez ?

J'utilise les k-moyennes (ou la segmentation hiérarchique).

Quelle(s) commande(s) tapez-vous pour obtenir cette segmentation ?

```
seeds.3groupes <- kmeans (seeds [, 1:7], centers = 3, iter.max = 30, nstart = 30)
```

ou si on fait une segmentation hiérarchique :

```
seeds.hclust <- hclust (dist (seeds [,-7]))
seeds.hclust.3groupes <- cutree (seeds.hclust, k = 3)
```

On veut comparer le résultat de cette segmentation avec les classes présentes dans le jeu de données.

Comment faites-vous cette comparaison graphiquement ?

On peut afficher deux fenêtres côte à côte : le graphique fait plus haut d'une part, un autre identique

où on utilise `seeds.3groupes$cluster` pour la couleur des points.

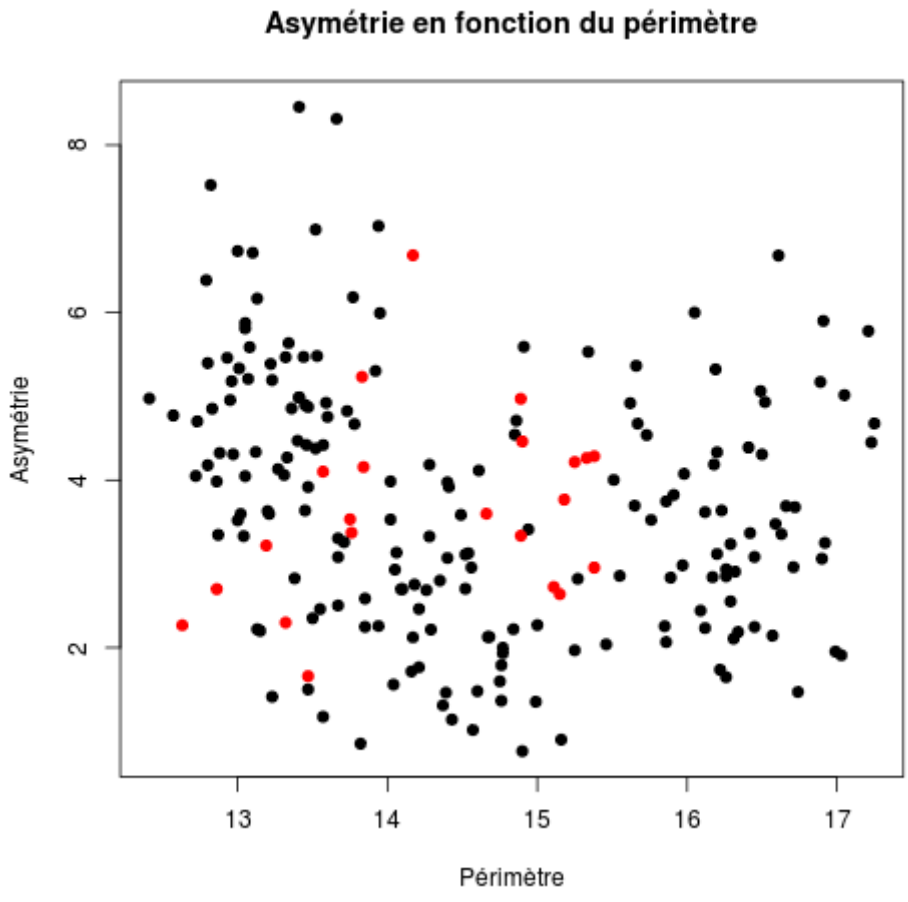
Cette solution est très approximative.

On peut faire mieux en affichant les points avec une couleur indiquant s'ils sont dans un groupe correspondant à leur classe ou pas. La petite difficulté est que la numérotation des groupes produits par `kmeans()` n'a pas de raison d'être la même que celle de l'attribut `variety`. Le plus simple est de comparer les graphiques précédents. À la question 5, on voit que le groupe le plus à gauche est le 3 (3 = couleur verte), celui du milieu est le 1 (1 = couleur noire) et celui de droite est le 2 (2 = couleur rouge). `kmeans()` a numéroté 3 le groupe de gauche (vert), celui du milieu 2/rouge et celui de droite 1/noir (vous pouvez avoir obtenu une autre numérotation).

On peut créer un vecteur indiquant les données se trouvant dans des segments différents dans les deux segmentations.

```
mal.classes <- rep (1, times = nrow (seeds))
mal.classes [which ((seeds$variety == 1) & (seeds.3groupes$cluster!= 2))] <- 2
mal.classes [which ((seeds$variety == 2) & (seeds.3groupes$cluster!= 1))] <- 2
mal.classes [which ((seeds$variety == 3) & (seeds.3groupes$cluster!= 3))] <- 2
plot (seeds [, 2], seeds [, 6], col = mal.classes,
      pch = ifelse (mal.classes == 1, 21, 19),
      main = "Asymétrie en fonction du périmètre",
      xlab = "Périmètre", ylab = "Asymétrie")
```

Donne la figure ci-dessous :



où les mal classés sont indiqués en rouge.

Pour une segmentation hiérarchique, le principe est exactement le même pour cette question et la suivante.

Comment faites-vous une table de confusion ?

```
table (seeds.3groupes$cluster, seeds$variety)
```

J'obtiens :

```
    1  2  3
1   1 60  9
2  60 10  0
3   0  2 68
```

Il y a donc 22 données qui ne sont pas dans le segment correspondant à leur classe.

Question 7. Comment effectuez-vous une ACP de ce jeu de données (sans considérer la classe) ?

```
seeds.acp <- prcomp (seeds [, -7], retx = T)
```

Question 8. Faire une segmentation en 3 groupes des points obtenus par l'ACP. Cette segmentation est-elle plus proche de la classe des données que la segmentation obtenue ci-dessus (expliquez) ?

```
seeds.acp.3groupes <- kmeans (seeds.acp$x, centers = 3, iter.max = 30, nstart =
30)
```

On peut constater que la segmentation est plus proche des classes en faisant une table de contingence :

```
table (seeds$variety, seeds.acp.3groupes$cluster)
```

ce qui donne :

```
    1  2  3
1   5  1 64
2   0 60 10
3  70  0  0
```

Il y a maintenant 16 données qui ne sont pas dans les mêmes segments.