TP 2

1. On considère μ_1 et μ_0 deux réels tel que $\mu_1 > \mu_0$. Soit (X, Y) un couple de variable aléatoire tel que Y suit une loi de Bernouilli de paramètre $p \in (0, 1)$ et

$$X|Y = 0 \sim \mathcal{N}(\mu_0, 1) \quad \text{et} \quad X|Y = 1 \sim \mathcal{N}(\mu_1, 1).$$

1. Compléter le code `R` ci-dessous afin que (x, y) soit une réalisation de (X, Y) et que la variable `pred` renvoie la valeur de $s^*(x)$ où s^* est le classifieur Bayes.
```
y <- rbinom(1,1,p)
x <- y*rnorm(...
pred <- ...
```
2. Écrire une fonction prenant en argument un entier n et renvoyant une estimation de $R(s^*)$
3. Pour estimer s^*, on se propose d'utiliser la fonction `lda` du package `MASS`
```
library(MASS)
help(lda)
```
Pour estimer les performances de cette méthode on propose d'utiliser le schéma suivant. On répète 100 fois les étapes suivantes
   (i) On simule un échantillon de taille n de même loi que (X, Y) sur lequel on construit l'estimateur $\hat{s}$ de s^*
   (ii) On simule un deuxième échantillon indépendant du précédent de taille 1000 sur lequel on calcule le risque empirique de $\hat{s}$ noté $\hat{R}$.

   Finalement, on calcule la moyenne (ainsi que l'écart type) des valeurs de $\hat{R}$ obtenues.

   (a) Compléter le code ci-dessous pour qu'il réalise les étapes (i) et (ii)
```
y.train <- rbinom(n,1,p)
x.train <- y*rnorm(...
model.lda <- lda(as.matrix(x.train), y.train)
y.test <- ...
x.test <- ...
pred.lda <- predict(model.lda, ...)   # voir help(predict.lda)
risque.estim <- mean(...)
```

   (b) En vous inspirant des question 3), 4), 5) de l'exercice 2) du TP 1, proposer un code qui permet de renvoyer une estimation du risque de classification de la méthode `lda` pour différentes valeurs de n, μ_1, μ_0 dans le cas $p = 0.5$. Comparer ces résultats au risque du classifieur de Bayes.

2. On se propose d'étudier les algorithmes de classification des `k plus proches voisins` et la `régression logistique` sur le jeu de données `iris`

1. Pour charger le jeu de données : `data(iris)`. À l'aide de la fonction `help`, décrire succinctement les données. En particulier, on précisera les variables explicatives et la variable à expliquer.
2. À l'aide de la fonction `summary` donner la moyenne et la médiane de chacune des variables explicatives. Vous préciserez aussi les écarts-types.
3. Que renvoie la commande `pairs(iris[,1:4])` ? Quelles variables semblent corrélées ? Donner le coefficient de corrélation des deux variables les plus corrélées.
4. On souhaite effectuer une ACP sur les variables explicatives. Pour cela, on effectue la commande suivante `AcpIris <- princomp(iris[,1:4], cor = T, scores =T)`
   (a) Que renseigne l'argument `cor = T` ?
   (b) Que revoie `AcpIris$sdev` ?
   (c) En déduire le pourcentage d'inertie expliqué par les deux premières composantes principales.
   (d) Que renvoie `AcpIris$scores` ?

(e) Donner le graphique de la représentation des individus dans le premier plan factoriel (on affectera une couleur pour chacune des trois classes. Que constatez-vous ?

(f) À l'aide de la fonction `biplot` déterminer la variable la plus corrélée à la première composante principale.
On peut optenir une meilleure sortie graphique en utilisant la fonction `autoplot` du package `ggfortify`. On pourra se référer à la page suivante.

(g) *bonus.* Donner un code `R` permettant de tracer le cercle des corrélations. Afficher alors le résultat obtenu.

5. On souhaite désormais construire une règle de classification permettant de discriminer les espèces `versicolor` et `virginica`. Pour l'algorithme des k plus proches voisins, on utilisera la fonction `knn` du package `class`. Pour la regression logistique, on utlisera la fonction `glm`.

(a) Construire un nouveau jeu de données, `iris2` ne contenant que les variables relatives aux deux espèces d'intérêt. Dans le nouveau jeu de données la variables à expliquer aura pour nom Y (on pourra utiliser la fonction `colnames`). De plus, on codera $Y = 1$ si l'espèce est `versicolor` et 0 sinon.
Que modifie la commande suivante `rownames(iris2) <- c()` ?

(b) Que renvoie les variables `pred1` et `pred2` définies ci-dessous ?
On précisera également ce que renseigne les argument `prob = F` et `family = binomial`.

```
N <- 80
idx1 <- sample(1:50, N/2, replace = F)
idx0 <- sample(51:100, N/2, replace = F)
dataL <- iris2[c(idx1,idx0),]
dataV <- iris2[-c(idx1,idx0),]
pred1 <- knn(train = dataL[,-5], test = dataV[,-5], cl = dataL[,5], k = 3, prob = F)
fit.glm <- glm(Y ~ ., data = dataL, family = binomial, maxit = 100)
pred2 <- predict(fit.glm, newdata = dataV[,-5], type = "response")
```

(c) Que renseigne la commande `maxit` de la fonction `glm`.

(d) En s'inspirant de la méthodologie proposée dans l'exercice 2 et de la question précédente, écrire un code permettant d'évaluer le risque de classification pour différentes valeurs de N, des algorithmes de classification des k plus proches voisins pour différentes valeurs de k, de la régression logistique, et de l'analyse discrimante de Fisher (`lda`). Discuter des résultats obtenus.

(e) *bonus.* En effectuant une régression logistique sur l'ensemble des individus du jeu de données `iris2`, déterminer la variable explicative la plus pertinente. Justifier votre réponse.