Créez vos propres fonctions R

Une fonction très simple en R ***Pour le cours #3***

Écrire la fonction dans un fichier texte (.txt), puis "sourcer" le fichier dans la console R

```
ecrire.nom = function( )
{
nom = ' Inscrivez votre nom ici '
cat('Mon nom est : ', nom, '\n')
# Commande équivalente : cat("Mon nom est : ", nom, "\n")
}

# Utilisation : ecrire.nom ( )
```

Une fonction qui réalise un calcul : additionner deux nombres

```
addition = function(a, b)
{
c = a+b
cat('c = ', c, '\n')
return(c)
}

# Utilisation :   toto = addition(32, 183)
                  toto
```

Variation sur le même thème : addition et soustraction de nombres

```
addition2 = function(a, b)
{
c = a+b
d = a-b
cat('c = ', c, '  d = ', d, '\n')
return(list(plus=c, moins=d))
}

# Utilisation :   toto2 = addition2(32, 183)
                  toto2
                  summary(toto2)
                  toto2$plus
                  toto2$moins
```

Écrivez sur une seule ligne une fonction qui calcule la longueur d'un vecteur

Écrivez sur une seule ligne une fonction qui normalise un vecteur (éq. 2.7)

Une fonction R pour l'analyse en composantes principales (ACP) ***Pour le cours #8***

Utilisez l'algèbre matricielle décrite aux pages 4-5 du fichier "Travaux_pratiques_en_R.pdf" pour créer votre propre fonction d'ACP. Le fichier de données aura déjà été lu dans la console R lors de l'utilisation de votre fonction. Il ne faut donc pas inclure la commande 'read.table' dans votre fonction.

Travaillez dans un fichier texte. Donnez d'abord un nom à votre fonction, comme nous l'avons fait dans les exercices précédents. Ouvrez l'accolade {. Recopiez ensuite les lignes de code en commençant à la ligne suivante, p. 4 :

```
Y.mat = as.matrix(Y)
```

jusqu'à la ligne suivante qui se trouve à la p. 5 :

```
G = F %*% diag(Y.eig$values^(-0.5))
```

Faites exécuter chaque ligne de code dans la console R pour bien comprendre son rôle. Réfléchissez à l'effet de chaque ligne de code dans l'algèbre de l'ACP étudiée au cours. Éliminez les lignes redondantes ou inutiles.

Pour le graphique, utilisez la fonction suivante : `biplot(F,U)`

Votre fonction devra produire la liste suivante d'éléments en sortie :
`return(list(eigen.val=Y.eig$values, U=U, F=F, Usc2=U2, G=G))`

Fermez l'accolade pour indiquer la fin de la fonction.

Lorsque vous aurez pu vérifier que la fonction de base tourne correctement, incorporez les deux options suivantes:

(1) Option: tableau **Y** centré seulement ou centré-réduit (**Y** doit être centré dans tous les cas)

```
# Le première ligne de la function doit contenir les informations suivantes:
nom.de.votre.fonction.acp = function(Y, scaling.type=1, standardize=FALSE)

# Modifiez la ligne de centrage pour incorporer l'option de centrage ou centrage-réduction:
Y.cent = scale(Y.mat, center=TRUE, scale=standardize)
```

(2) Option pour cadrage de type 1 ou de type 2

```
if(scaling.type == 1) {
  # Que faut-il faire si l'expression logique est vraie?
  biplot(F,U)
  } else {
  # Que faut-il faire si l'expression logique est fausse?
  biplot(G,U2)
  }
```