Correction du TP en R

# Exercice 1 : Utilisation des fonctions mathématiques

Pour cet exercice, il s'agit simplement d'appeler les fonctions mathématiques en R pour obtenir les résultats correspondants. Voici comment j'ai procédé :

# On appelle la fonction sin pour obtenir le sinus de 30  
sin(30)  
  
# On appelle la fonction tan pour obtenir la tangente de 30  
tan(30)  
  
# On appelle la fonction log pour obtenir le logarithme naturel de 10  
log(10)  
  
# On appelle la fonction round pour arrondir la valeur de pi à deux décimales  
round(3.14159, 2)  
  
# On appelle la fonction cos pour obtenir le cosinus de 30  
cos(30)

# Exercice 2 : Programme interactif en R

Pour cet exercice, j'ai développé un programme interactif en R. Voici la méthode que j'ai suivie :  
  
1. J'ai d'abord demandé à l'utilisateur de saisir son prénom.  
2. Ensuite, j'ai affiché un message de bienvenue.  
3. J'ai demandé une série de notes (5 au maximum).  
4. J'ai calculé la somme et la moyenne de ces notes, puis affiché les résultats.

# Lire le prénom de l'utilisateur  
prenom <- readline(prompt = "Entrez votre prénom: ")  
  
# Afficher le message de bienvenue  
cat("Bienvenue,", prenom, "  
")  
  
# Lire une série de notes  
notes <- scan(what = numeric(), nmax = 5, prompt = "Entrez vos notes (5 maximum) : ")  
  
# Calcul de la somme des notes  
somme\_notes <- sum(notes)  
  
# Calcul de la moyenne des notes  
moyenne\_notes <- mean(notes)  
  
# Affichage des résultats  
cat("La somme des notes est :", somme\_notes, "  
")  
cat("La moyenne des notes est :", moyenne\_notes, "  
")

# Exercice 3 : Application de gestion de stock

Ici, j'ai créé un programme qui demande la quantité de produits commandés, puis qui :  
  
1. Détermine si c'est une petite, moyenne ou grande commande.  
2. Vérifie si la commande peut être divisée en paquets égaux.

# Demander la quantité de produits commandés  
quantite <- as.integer(readline(prompt = "Entrez la quantité commandée : "))  
  
# Déterminer le type de commande  
if (quantite < 10) {  
 cat("Petite commande  
")  
} else if (quantite <= 50) {  
 cat("Commande moyenne  
")  
} else {  
 cat("Grande commande  
")  
}  
  
# Vérifier si la commande est divisible en paquets égaux  
if (quantite %% 2 == 0) {  
 cat("La commande peut être divisée en paquets égaux.  
")  
} else {  
 cat("La commande ne peut pas être divisée en paquets égaux.  
")  
}

# Exercice 4 : Analyse des ventes mensuelles

Pour cet exercice, j'ai créé un programme qui :  
  
1. Demande les ventes mensuelles pendant 6 mois.  
2. Affiche les mois où les ventes sont inférieures à 3000 dh.  
3. Vérifie si les ventes totales atteignent un objectif annuel de 20000 dh. Si l'objectif n'est pas atteint, le programme augmente les ventes du dernier mois de 10% jusqu'à ce que l'objectif soit atteint.

# Entrer les ventes pour 6 mois  
ventes <- numeric(6)  
for (i in 1:6) {  
 repeat {  
 ventes[i] <- as.numeric(readline(prompt = paste("Entrez les ventes du mois", i, ": ")))  
 if (ventes[i] > 0) break  
 }  
}  
  
# Afficher les mois avec des ventes inférieures à 3000 dh  
for (i in 1:6) {  
 if (ventes[i] < 3000) {  
 cat("Les ventes du mois", i, "sont inférieures à 3000 dh :", ventes[i], "  
")  
 }  
}  
  
# Vérifier si l'objectif annuel est atteint  
objectif <- 20000  
total\_ventes <- sum(ventes)  
  
while (total\_ventes < objectif) {  
 ventes[6] <- ventes[6] \* 1.10 # Augmenter de 10% les ventes du dernier mois  
 total\_ventes <- sum(ventes)  
}  
  
cat("L'objectif de ventes annuel est atteint avec un total de :", total\_ventes, "  
")

# Exercice 5 : Fonctions en R

Pour cet exercice, j'ai créé quatre fonctions différentes :  
  
1. Une fonction qui n'accepte aucun argument et affiche un message de bienvenue.  
2. Une fonction qui calcule la somme de deux nombres.  
3. Une fonction qui calcule la somme de trois nombres avec des valeurs par défaut.  
4. Une fonction qui calcule la moyenne d'une série de nombres.

# Fonction de bienvenue  
bienvenue <- function() {  
 cat("Bienvenue dans ce programme R  
")  
}  
  
# Fonction de somme de deux nombres  
somme\_deux <- function(a, b) {  
 return(a + b)  
}  
  
# Fonction de somme de trois nombres avec valeurs par défaut  
somme\_trois <- function(a = 1, b = 2, c = 3) {  
 return(a + b + c)  
}  
  
# Fonction de calcul de la moyenne  
moyenne <- function(...) {  
 return(mean(c(...)))  
}  
  
# Appel des fonctions  
bienvenue()  
cat("Somme de 2 et 3 :", somme\_deux(2, 3), "  
")  
cat("Somme de 1, 2, 3 (valeurs par défaut) :", somme\_trois(), "  
")  
cat("Moyenne de 1, 2, 3, 4 :", moyenne(1, 2, 3, 4), "  
")