

Exercices de programmation en R – prérequis

Programme couvert :

Assignation de variables, utilisation des fonctions built-in de R.

Les vecteurs et leur manipulation.

=====

Avant toute chose :

Les exercices proposés sont donnés dans un ordre progressif qui suit celui du diaporama associé. Ils ont des niveaux de difficulté variable. Pour chaque exercice, des fonctions R vous sont suggérées. N'hésitez pas à consulter leur menu d'aide en tapant soit « ?nom_fonction », soit « help(nom_fonction) ». Vous pouvez parvenir aux mêmes résultats avec d'autres commandes. Les commandes les plus courtes sont en général les meilleures. Nous vous fournirons les corrections détaillées.

1. Utilisez un éditeur de texte avec coloration syntaxique pour votre code (éditeur dans Rstudio, ou Tinn-R ou notepad++ par exemple). Sauvegardez vos commandes dans un fichier mes_commandes.R dans le répertoire de votre choix. Exécutez les commandes une par une dans votre console. Vous pouvez ajouter des commentaires qui ne seront pas exécutés si vous les précédez du caractère « # ».

2. Définissez votre espace de travail

↳ Fonction à utiliser : `setwd()`

3. Identifiez la version R de votre environnement et packages installés.

↳ Fonction à utiliser : `sessionInfo()`

4. A la fin, sauvegardez votre session pour la prochaine fois et l'historique !

↳ Fonctions à utiliser : `save.image()`, `paste()`, `Sys.Date()`,
`savehistory()`

5. Lors de votre nouvelle session, si vous en avez besoin, rechargez les objets R précédemment sauvegardés.

↳ Fonction à utiliser : `load()`

6. Vous pouvez élargir la largeur d'affichage :

↳ Fonction à utiliser, par exemple pour 160 : `options(width="160")`

Exercices:**▪ Exercice 1 :**

Créez le vecteur « vec1 » contenant la suite des entiers de 1 à 12. Ajoutez à la fin de ce vecteur les valeurs 16, 17, 18.

↳ Fonctions à utiliser : ":" et c()

▪ Exercice 2 :

Créez le vecteur « vec2 » contenant les valeurs suivantes : 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0. Réalisez cette action de différentes manières.

↳ Fonctions à utiliser : c(), seq()

▪ Exercice 3 :

Effectuez l'opération suivante : $4850 / 26$. Affichez le résultat avec seulement 2 ou 3 décimales. Existe-t-il d'autres fonctions R permettant de réaliser une réduction de décimales ?

↳ Fonctions à utiliser : round(), help()

▪ Exercice 4 :

Créez le vecteur « vec3 » contenant les noms suivants : « individu1 », « individu2 », ... , « individu100 ». Attention au séparateur utilisé !

↳ Fonctions à utiliser : paste()

▪ Exercice 5:

Affichez la date du jour. Sauvegardez le résultat dans une variable « dateJour ».

Créez la variable « m1 » dans laquelle vous assignerez la phrase « Je me souviendrai longtemps de mon premier cours de R, le ».

Combinez les variables dateJour et m1 afin d'afficher à l'écran la phrase suivante « Je me souviendrai longtemps de mon premier cours de R, le 2020-03-07 ».

Comptez le nombre de caractères de cette phrase.

↳ Fonctions à utiliser : Sys.Date(), paste(), nchar()

▪ Exercice 6:

Créez la variable « vec4 » dans laquelle vous assignerez les 24 lettres de l'alphabet en lettres capitales. Remplacez ensuite par des lettres minuscules dans le vecteur vec4 toutes les lettres d'indice impair. Inversez l'ordre du vecteur vec4. Affichez le vecteur vec4 ainsi obtenu.

↳ Fonctions à utiliser : `LETTERS`, `seq()`, `tolower()`, `rev()`

▪ Exercice 7:

Créez un vecteur vec5 dans lequel vous entrerez les entiers de 1 à 10, en entrant d'abord les valeurs impaires puis les valeurs paires. Ajoutez ensuite, sous forme d'attribut, le nom « impair » ou « pair » à chacune des valeurs de ce vecteur. Réordonnez le vecteur v5 dans l'ordre croissant des valeurs. Quel est le type de données du vecteur vec5 ?

Affichez à présent les 2^{ème}, 4^{ème}, 9^{ème}, 8^{ème} et 7^{ème} valeurs. Remplacez-les ensuite dans le vecteur vec5 par les lettres de l'alphabet du vecteur vec4 de même indice. Affichez le vecteur vec5 ainsi obtenu. Quelle est maintenant le type de données du vecteur vec5 ?

↳ Fonctions à utiliser : `c()`, `seq()`, `sort()`, `mode()`

▪ Exercice 8:

Assignez dans un vecteur vec6 100 nombres entiers tirés aléatoirement dans un ensemble de valeurs comprises entre 1 et 1000. Calculez la moyenne, la médiane, l'écart-type et les quantiles de cette série de 100 valeurs. Retirez les 50 dernières valeurs du vecteur vec6.

↳ Fonctions à utiliser : `sample()`, `mean()`, `median()`, `sd()`, `quantile()`, `summary()`

▪ Exercice 9:

Créez un vecteur v7 dans lequel vous assignerez 50 valeurs tirées aléatoirement parmi 50 données manquantes et 50 données issues d'une loi normale de moyenne 3 et de variance 4. Calculez la moyenne du vecteur vec7. Comptez le nombre de données manquantes dans le vecteur vec7.

Dans un vecteur vec8, gardez uniquement les valeurs positives du vecteur vec7. Affichez le vecteur vec8 pour vérifier qu'il ne contient que des données positives.

↳ Fonctions à utiliser : `sample()`, `c()`, `rep()`, `rnorm()`, `mean()` avec l'argument `na.rm=TRUE`, `is.na()`, `sum()`, `which()`

▪ Exercice 10:

Simulez 100 lancers d'une pièce de monnaie. Pour cela, réalisez un tirage avec remise dans un ensemble de deux valeurs possibles (« pile » et « face ») et assignez le résultat dans un vecteur `results`. Comptez le nombre de piles et de faces.

↳ Fonctions à utiliser : `sample()`, `table()`

▪ Exercice 11:

Réalisez la même simulation que précédemment (exercice 10), mais en biaisant la pièce de monnaie (la probabilité d'obtenir « pile » doit seulement être de 0,3). Quelles proportions de piles et de faces avez-vous obtenues ? Recodez tous les piles par les valeurs numériques 0 et les faces par 1. Comptez le nombre de faces.

↳ Fonctions à utiliser : `sample()`, `table()`, `which()`, `as.numeric()`, `sum()`

▪ Exercice 12:

Importer dans votre session R le jeu de données « `precip` ». Ce jeu de données est pré-enregistré dans le logiciel R et regroupe des données de précipitations dans différentes villes américaines.

Stocker dans le vecteur « `villes` » la liste des villes pour lesquelles des mesures sont disponibles.

Combien y en a-t-il ? Afficher le niveau de précipitation des villes suivantes : Philadelphia, Columbia, Baltimore, Sacramento.

↳ Fonctions à utiliser : `data(precip)`, `names()`, `length()`