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 avec des commentaires supplémentaires **« Pour aller plus loin »**.

1. Utilisez un editeur de texte avec coloration syntaxique pour votre code (editeur 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 éxécutés si vous les précédez du caractère « # ».

2. Définissez votre espace de travail

   ↳ `Fonction recommandées : setwd()`

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

   ↳ `Fonction recommandées : sessionInfo()`

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

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

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

   ↳ `Fonction recommandées : load()`

6. Vous pouvez élargir la largeur d'affichage :

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

Exercices:

==========

Exercice 1 : création de vecteurs numériques

- 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. Refaites ces opérations en une seule ligne de commande et affichez ensuite le vecteur `vec1`.

  ↳ `Fonctions recommandées : ":" et c()`

- 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 recommandées : c(), seq()`

Exercice 2 : manipulation de données numériques

- 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 recommandées : round(), help()`

Exercice 3 : création d'un vecteur avec des chaînes de caractères incrémentées

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

  ↳ `Fonctions recommandées : paste()`

Exercice 4: création et manipulation de vecteurs de chaînes de caractères

- 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 YYYY-MM-DD».
- Comptez le nombre de caractères de cette phrase.

  ↳ `Fonctions recommandées : Sys.Date(), paste(), nchar()`

Exercice 5: utilisation d'une constante de R et manipulation de chaînes de caractères

- 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'odre du vecteur vec4. Affichez le vecteur vec4 ainsi obtenu.

  ↳ `Fonctions et constantes recommandées : LETTERS, seq(), tolower(), rev()`

Exercice 6: manipulation avancée d'un vecteur

- 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 2ème, 4ème, 9ème , 8ème et 7ème 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.
- Quel est nouveau le type de données du vecteur vec5 ?

  ↳ `Fonctions recommandées : c(), seq(), sort(), mode()`

Exercice 7: tirage aléatoire et statistiques descriptives de données numériques

- Tirez aléatoirement 10 nombres dans un ensemble de valeurs comprises entre 1 et 100.

  ↳ `Fonction recommandée : sample()`
- Assignez dans un vecteur « vec6 » 100 nombres entiers tirés aléatoirement dans un ensemble de valeurs comprises entre 1 et 1000.
- Constatez bien que le vecteur `vec6` a été généré par un nouvel échantillonnage et ne contient pas (ou avec une très faible probabilité) les mêmes valeurs que celles obtenues juste au-dessus sans assigner les valeurs dans une variable.
- Calculer la moyenne, la médiane, l'écart-type et les quartiles de cette série de 100 valeurs.
- Retirez les 50 dernières valeurs du vecteur vec6.

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

Exercice 8: tirage aléatoire d'une loi de probabilité connue et filtrage de valeurs

- 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.

- Calculer 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 recommandées : sample(),c(), rep(), rnorm(), mean() avec l'argument na.rm=TRUE, is.na(), sum(), which()`

- 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 recommandées : which()`

Exercice 9: comptage de valeurs, remplacement de valeurs et coercion

- 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 recommandées : sample(), table()`

- Réalisez la même simulation que ci-dessus, 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 recommandées : sample(), table(), which(), as.numeric(), sum()`

Exercice 10: importation et filtrage de données

- 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 recommandées : data(precip), names( ), length( )`

==========

Crédits : remerciements à Leslie Regad et Gaëlle Lelandais pour plusieurs exercices