

Travaux pratiques

Master 1 Actuariat ISF

Ibrahima SY

10/20/2021

Partie II : Manipulation de vecteurs

Considérons le vecteur suivant :

$$x = [1, 2, 3, 4, 5]$$

1. Créer ce vecteur dans R et le stocker dans un objet que l'on appellera `xx`
2. Afficher le mode de `x`, puis sa longueur ;
3. Extraire le premier élément, puis le dernier ;
4. Extraire les trois premiers éléments et les stocker dans un vecteur que l'on nommera `a` ;
5. Extraire les éléments en position 1, 3, 5 ; les stocker dans un vecteur que l'on nommera `b` ;
6. Additionner le nombre 10 au vecteur `x`, puis multiplier le résultat par 2 ;
7. Effectuer l'addition de `a` et `b`, commenter le résultat ;
8. Effectuer l'addition suivante : `x+a`, commenter le résultat, puis regarder le résultat de `a+x` ;
9. Multiplier le vecteur par le scalaire `c` que l'on fixera à 2 ;
10. Effectuer la multiplication de `a` et `b`, commenter le résultat ;
11. Effectuer la multiplication suivante : `x*a`, commenter le résultat ;
12. Récupérer les positions des multiples de 2 et les stocker dans un vecteur que l'on nommera `ind`, puis conserver uniquement les multiples de 2 de `x` dans un vecteur que l'on nommera `mult_2` ;
13. Afficher les éléments de `x` qui sont multiples de 3 *et* multiples de 2 ;
14. Afficher les éléments de `x` qui sont multiples de 3 *ou* multiples de 2 ;
15. Calculer la somme des éléments de `x` ;
16. Remplacer le premier élément de `x` par un 4 ;
17. Remplacer le premier élément de `x` par la valeur `NA`, puis calculer la somme des éléments de `x` ;
18. Lister les objets en mémoire dans la session R ;
19. Supprimer le vecteur ;
20. Supprimer la totalité des objets de la session.

Partie II : Manipulation de listes

1. évaluer le code suivant : `TRUE+FALSE+TRUE*4` et le commenter ;
2. évaluer les expressions suivantes : `c(1, 4, TRUE)`, et `c(1, 4, TRUE, "bonjour")`, commenter ;
3. Créer une liste que l'on appellera `l` et qui contient les éléments 1, 4 et `TRUE` en première, deuxième et troisième positions, respectivement ;
4. Extraire le premier élément de la liste `l`, et afficher son mode. En faire de même avec le troisième élément, et commenter ;
5. Ajouter un quatrième élément à la liste `l` : `"bonjour"`, puis afficher la structure de `l` ;
6. Retirer le troisième élément de la liste `l` ;

7. Créer une liste de trois éléments : votre nom, votre prénom, et votre année de naissance. Ces trois éléments de la liste devront être nommés respectivement "nom", "prenom" et "année de naissance". Stocker la liste ainsi créée dans un objet nommé `moi` ;
8. Extraire le prénom de la liste `moi` de deux manières : en utilisant l'indice, et en utilisant le nommage ;
9. Créer une liste avec la même structure que celle de `moi`, en la remplissant avec les informations d'une autre personne et la nommer `toi`. Puis, créer la liste `personnes`, qui contiendra les listes `toi` et `moi` ;
10. Extraire la liste `toi` de `personnes` (en première position) ; 11 Extraire directement depuis `personne` le prénom de l'élément en première position.

Partie III : Manipulation de données

1. À l'aide de la fonction `read_excel()` du *package* `readr`, importer le contenu de la feuille intitulée `notes_2012` du fichier Excel disponible à l'adresse suivante : http://egallic.fr/Enseignement/R/Exercices/donnees/notes_etudiants.xlsx et le stocker dans une variable que l'on nommera `notes_2012` ;
2. Afficher les 6 premières lignes du jeu de données, puis les dimensions du tableau ;
3. Conserver uniquement la colonne `note_stat` du tableau de données `notes_2012` dans un objet que l'on appellera `tmp` ;
4. Conserver uniquement les colonnes `num_etudiant`, `note_stat` et `note_macro` dans l'objet `tmp` ;
5. Remplacer le contenu de `tmp` par les observations de `notes_2012` pour lesquelles l'individu a obtenu une note de stat supérieure (strictement) à 10 ;
6. Remplacer le contenu de `tmp` par les observations de `notes_2012` pour lesquelles l'individu a obtenu une note comprise dans l'intervalle (10,15)(10,15) ;
7. Regarder s'il y a des doublons dans le tableau de données `notes_2012` ; le cas échéant, les retirer du tableau ;
8. Afficher le type des données de la colonne `num_etudiant`, puis afficher le type de toutes les colonnes de `notes_2012` ;
9. Ajouter au tableau `notes_2012` les colonnes suivantes :
 - `note_stat_maj` : la note de stat (`note_stat`) majorée d'un point,
 - `note_macro_maj` : la note de macro (`note_macro`) majorée de trois points (le faire en deux étapes : d'abord deux points en plus, puis un point) ;
10. Renommer la colonne `year` en `annee` ;
11. Depuis le fichier `notes_etudiants.xlsx` (c.f. question 1), importer le contenu des feuilles `notes_2013`, `notes_2014` et `prenoms` et le stocker dans les objets `notes_2013`, `notes_2014` et `prenoms` respectivement ;
12. Empiler le contenu des tableaux de données `notes_2012`, `notes_2013` et `notes_2014` dans un objet que l'on nommera `notes` ;
13. Fusionner les tableaux `notes` et `prenoms` à l'aide d'une jointure gauche, de manière à rajouter les informations contenues dans le tableau `prenoms` aux observations de `notes`. La jointure doit se faire par le numéro d'étudiant et l'année, l'objet final viendra remplacer le contenu de `notes` ;
14. Trier le tableau `notes` par années croissantes et notes de macro décroissantes ;
15. Changer le type des colonnes `annee` et `sexe` en facteur ;
16. Créer une colonne `apres_2012` qui prend la valeur `TRUE` si l'observation concerne une note attribuée après 2012 ;
17. à l'aide de la fonction `summarize()` du *package* `{dplyr}`, calculer :
 - la moyenne et l'écart-type annuels des notes pour chacune des deux matières,
 - la moyenne et l'écart-type annuels et par sexe des notes pour chacune des deux matières ;
18. En utilisant la fonction `pivot_longer()` du *package* `{tidyr}`, créer un tableau dans lequel chaque ligne renseigne le numéro d'étudiant, l'année, le prénom, le sexe, l'enseignement (macro ou stat) et la note ;
19. En repartant de l'objet obtenu à la question précédente, utiliser la fonction `pivot_wider()` du *package* `{tidyr}` pour retomber sur le même tableau que `notes`.

Partie IV : Manipulation de tableaux de chaînes de caractères

1. Créer les objets **a** et **b** afin qu'il contiennent respectivement les chaînes de caractères suivantes : 23 à 0 et C'est la piquette, Jack ! ;
2. Créer le vecteur **phrases** de longueur 2, dont les deux éléments sont **a** et **b** ;
3. À l'aide de la fonction appropriée dans le *package* **{stringr}**, afficher le nombre de caractères de **a**, de **b**, puis appliquer la même fonction à l'objet **phrases** ;
4. En utilisant la fonction **str_c()**, concaténer **a** et **b** dans une seule chaîne de caractères, en choisissant la virgule comme caractère de séparation ;
5. Concaténer les deux éléments du vecteur **phrases** en une seule chaîne de caractères, en les séparant par le caractère de retour à la ligne, puis utiliser la fonction **cat()** pour afficher le résultat dans la console ;
6. Appliquer la même fonction que dans la question précédente à l'objet suivant : **c(NA, phrases)** et commenter ;
7. Mettre en majuscules, puis en minuscules les chaînes du vecteur **phrases** (afficher le résultat, ne pas modifier **phrases**) ;
8. À l'aide de la fonction **word()** du *package* **{stringr}**, extraire le mot **la**, puis **Jack** de la chaîne **b** ;
9. Même question que la précédente, en utilisant la fonction **str_sub()** ;
10. À l'aide de la fonction **str_detect()**, rechercher si le motif **piqu** puis **mauvais** sont présents dans **b** ;
11. À l'aide de la fonction **str_detect()**, rechercher si le motif **piqu** est présent dans les éléments du vecteur **phrases** ;
12. À l'aide de la fonction **str_detect()**, rechercher si le motif **piqu** ou le motif **à** sont présents dans les éléments du vecteur **phrases** ;
13. En utilisant la fonction **str_locate()**, retourner les positions de la première occurrence du caractère **a** dans la chaîne **b**, puis essayer avec le caractère **w** pour observer le résultat retourné ;
14. Retourner toutes les positions du motif **a** dans la chaîne **b** ;
15. En utilisant la fonction **str_replace()**, remplacer la première occurrence du motif **a**, par le motif **Z** (afficher le résultat, ne pas modifier **phrases**) ;
16. Remplacer toutes les occurrences de **a** par **Z** dans la chaîne **b** (afficher le résultat, ne pas modifier **phrases**) ;
17. Utiliser la fonction **str_split()** pour séparer la chaîne **b** en utilisant la virgule comme séparateur de sous-chaînes ;
18. Retirer tous les caractères de ponctuation de la chaîne **b**, puis utiliser la fonction **str_trim()** sur le résultat pour retirer les caractères blancs du début et de la fin de la chaîne.

Partie V : Manipulation de tableaux de dates

1. En utilisant la fonction **as.Date()**, stocker la date du 29 août 2015 dans un objet que l'on appellera **d** puis afficher la classe de l'objet ;
2. À l'aide de la fonction appropriée, afficher la date du jour ;
3. À l'aide de la fonction **as.Date()**, stocker sous forme de date la chaîne de caractères suivante : 29-08-2015 ;
4. Utiliser les fonctions **as.POSIXct()** et **as.POSIXlt** pour stocker la chaîne de caractères 2015-08-29 20:30:56 sous forme de dates dans des objets nommés **d_ct** et **d_lt** respectivement ; utiliser ensuite la fonction **unclass()** sur les deux objets pour comparer la façon dont R a stocké l'information ;
5. Utiliser la fonction appropriée du *package* **{lubridate}** pour stocker la chaîne de caractères 2015-08-29 sous forme de date ;
6. Même question avec la chaîne 2015-08-29 20:30:56 ;
7. Utiliser la fonction **ymd_hms()** pour stocker la date et l'heure actuelle, en précisant le fuseau horaire, puis afficher la date et l'heure correspondantes à New York City ;
8. Considérons le vecteur **x** :

```
x <- c(ymd_hms("2015-08-29 20:30:56", tz = "Europe/Paris"), ymd_hms("2015-09-15  
08:10:33", tz = "Europe/Paris"))
```

Extraire l'année, le mois, le jour, les heures, les minutes et les secondes du premier élément de **x** à l'aide des fonctions appropriées du *package* **{lubridate}** ;

9. Appliquer les mêmes fonctions au vecteur **x** ;
10. Au premier élément de **x**, ajouter :
 - une seconde,
 - un jour,
 - un mois
 - deux années ;
1. Tester si la date du premier élément de **x** vient avant celle du second élément ;
2. En utilisant la fonction **interval()** du *package* **{lubridate}**, créer un intervalle de dates entre les deux éléments de **x**, puis afficher le nombre de jours, puis le nombre d'heures, puis le nombre d'années séparant les deux dates ;
3. En utilisant la fonction **seq()**, créer une séquence de dates avec un intervalle de 5 jours entre chaque date, commençant à la date du premier élément de **x** et se terminant à la date du second élément de **x** (la séquence sera tronquée avant) ;
4. Convertir en date les deux chaînes de caractères suivantes : **Sam 29 Août 2015** et **Sat 29 Aug 2015**.