

Exercise Set Number 2

Feryal WINDAL

Spetember 2023

Summary of Instructions

Commande	Description
<code>rev(sort(s))</code>	Trie les coordonnées d'un vecteur par ordre décroissant.
<code>s[I]</code>	Crée une suite composée des éléments de la suite <code>s</code> indexées par <code>I</code> . Ici <code>I</code> peut être de la forme : <code>I</code> est entier. Exemple : <code>s[3]</code> renvoie le 3ème élément de <code>s</code> . <code>I</code> est une suite. Exemple : <code>s[3:5]</code> renvoie les 3ème, ..., 5ème éléments de <code>s</code> . <code>I</code> est une condition. Exemple : <code>s[t>3]</code> renvoie les éléments de <code>s</code> correspondants aux éléments de <code>t</code> qui sont supérieurs à 3.
<code>s[-I]</code>	Crée une suite composée des éléments de <code>s</code> qui sont complémentaires à ceux indexés par <code>I</code> .
<code>tail(s,n=k)</code>	Renvoie les <code>k</code> derniers éléments ou les dernières lignes de l'objet <code>s</code> .
<code>nrow(s)</code>	Donne le nombre de lignes de l'objet <code>s</code> .
<code>ncol(s)</code>	Donne le nombre de colonnes de l'objet <code>s</code> .
<code>tab\$col</code>	Renvoie le vecteur composé des éléments de la colonne <code>col</code> de la table <code>tab</code> . Exemple : <code>e<-amis\$email</code> initialise la variable <code>e</code> avec les valeurs de la colonne <code>email</code> de la table <code>amis</code> .
<code>rbind(r1,...,rk)</code>	Crée une matrice ou un tableau de données en collant les <code>r1, ..., rk</code> l'un en dessous de l'autre.
<code>cbind(c1,...,ck)</code>	Crée une matrice ou un tableau de données en collant les <code>c1, ..., ck</code> l'un à côté de l'autre.
<code>scan()</code>	Saisit « au clavier » un jeu de données numériques.
<code>getwd()</code>	Affiche le répertoire de travail dans la console.
<code>setwd(Chemin)</code>	Définit <code>Chemin</code> comme répertoire de travail.
<code>file.choose()</code>	Permet de sélectionner facilement un fichier stocké sur l'ordinateur avec une boîte de dialogue conviviale.
<code>load("file")</code>	Charge le contenu du fichier de sauvegarde « <code>file</code> » au format R en mémoire.
<code>save(objet,file="file")</code>	Sauvegarde l'objet <code>objet</code> dans le fichier de sauvegarde « <code>file</code> ».

Commande	Description
<code>read.table("file")</code>	Lit le fichier de données <code>file</code> ne contenant pas les noms des variables en première ligne.
<code>read.table("file",header=T)</code>	Lit le fichier de données <code>file</code> en spécifiant que les noms des variables sont contenus dans la première ligne. Exemple : <code>t<-read.table("table.txt",header=T).</code>
<code>write.table(objet,"file")</code>	Écrit objet dans le fichier de données <code>file</code> .
<code>read.csv("file")</code>	Lit un fichier de données <code>file</code> au format CSV anglo-saxon.
<code>write.csv(objet,"file")</code>	Écrit objet dans un fichier de données <code>file</code> au format CSV anglo-saxon.
<code>read.csv2("file")</code>	Lit un fichier de données <code>file</code> au format CSV français.
<code>write.csv2(objet,"file")</code>	Écrit objet dans un fichier de données <code>file</code> au format CSV français.
Bibliothèque <code>xlsx</code> <code>read.xlsx("file",sheetIndex)</code>	Environnement Windows, MacOS X ou Linux Lit la feuille de calculs numéro <code>sheetIndex</code> du fichier de données <code>file</code> au format Excel.
Bibliothèque <code>xlsx</code> <code>write.xlsx(objet,"file",sheetName)</code>	Environnement Windows, MacOS X ou Linux Écrit objet dans la feuille de calculs <code>sheetName</code> du fichier de données <code>file</code> au format Excel.
Bibliothèque <code>gdata</code> <code>read.xls("file")</code>	Environnement Windows, MacOS X ou Linux Lit un fichier de données <code>file</code> au format Excel.

Commande	Description
<code>s<-valeur</code> <code>s=valeur</code>	Affecte valeur à l'objet <code>s</code> .
<code>rm(s)</code>	Supprime l'objet <code>s</code> .
<code>mode(s)</code>	Affiche le mode (numérique, caractère,...) de l'objet <code>s</code> .
<code>class(s)</code>	Affiche la classe (<code>data.frame</code> ,...) de l'objet <code>s</code> .
<code>names(s)</code>	Renvoie les noms des éléments de l'objet <code>s</code> . Si <code>s</code> est un tableau de données, renvoie les noms des colonnes.
<code>length(s)</code>	Affiche le nombre d'éléments contenus dans l'objet <code>s</code> .
<code>dim(s)</code>	Donne les dimensions de l'objet <code>s</code> .
<code>as.character(s)</code>	Transforme l'objet <code>s</code> en caractères.
<code>as.list(s)</code>	Transforme l'objet <code>s</code> en liste.
<code>as.logical(s)</code>	Transforme l'objet <code>s</code> en booléen.
<code>as.numeric(s)</code>	Transforme l'objet <code>s</code> en numérique.
<code>is.character(s)</code>	Teste si l'objet <code>s</code> est un caractère.
<code>is.list(s)</code>	Teste si l'objet <code>s</code> est une liste.
<code>is.logical(s)</code>	Teste si l'objet <code>s</code> est un booléen.
<code>is.na(s)</code>	Teste si l'objet <code>s</code> a des valeurs manquantes.
<code>is.numeric(s)</code>	Teste si l'objet <code>s</code> est numérique.
<code>is.null(s)</code>	Teste si l'objet <code>s</code> est nul.
<code>c(s₁, s₂, ..., s_k)</code>	Crée un vecteur en collant les <code>s₁, s₂, ..., s_k</code> dans l'ordre. Exemple : <code>c(2:5,7,seq(8,9,0.5))</code> produit 2,3,4,5,7,8,8.5,9.
<code>sort(s)</code>	Trie les coordonnées d'un vecteur par ordre croissant.

Construction de vecteurs

1. Create the vector `v` equal to (101 ; 102 ;... ; 112).
2. Create the vector `w` of length 12 formed by four times the sequence of numbers (4 ; 6 ; 3).
3. Create the vector `x` consisting of eight times the number 4, seven times the number 6 and five times the number 3.

More vectors

1. Enter the weight variable containing the following 15 values : 28 ; 27, 5 ; 27 ; 28 ; 30, 5 ; 30 ; 31 ; 29, 5 ; 30 ; 31 ; 31 ; 31, 5 ; 32 ; 30 ; 30, 5.
2. Enter the variable weight1 containing the following 5 values : 40 ; 39 ; 41 ; 37, 5 ; 43.
3. Without redoing any entry, create the variable new.weight containing 20 values : - the five values of weight1 repeated twice - and the last ten values of weight.
4. Save, in your working directory, the new.weight variable in a sheet named "New Weight" of the Excel workbook "Weight.xls".

Imagination exercise

1. Choose the names of 10 people and create the name vector containing them.
2. Create the vector or age data table containing the ages of the previous 10 people (between 20 and 60) and use the names of the people as the caption for the age vector or as the row name for the age data table.
3. Create the vector or weight data table containing the weight of the 10 people (between 50 and 100 kg) again using the names of the people as the legend for this vector or as the name of the rows for this data table.
4. Create the vector or size data table containing the height of the 10 people (between 1m60 and 2m) again using the names of the people as the caption for this vector or as the row name for this data table.
5. Create the weight.lourds vector containing the weight of people over 80 kg, again using the names of the people as the caption for this vector.
6. Create the height.weight.high vector containing the height of people over 80 kg, making sure to keep the name associated with each weight.
7. Create the height.old.weight.high vector containing the height of people over 80 kg and over 30 years old, making sure to keep the name associated with each height. To answer this question, you can use the AND logical connector whose syntax is given in the problem below or in the help on logical operators accessible by typing the instruction `?Logic`.

Logical operators

Extracting or replacing values in a vector involves the use of logical vectors.

1. Build the sequence vector which is a sequence of integers going from 1 to 12.
2. You want to know the elements of the sequence vector strictly greater than 6. What is the command line that you must type to obtain the result ?
3. You want to know the elements of the sequence vector strictly less than 6. What is the command line that you must type to obtain the result ?

4. You want to know the elements of the sequence vector equal to 6. What is the command line that you must type to obtain the result ?
5. You want to know if the first element of the sequence vector is less than or equal to 8 and greater than or equal to 4. What is the command line you must type to get the result ?
6. You want to know the elements of the sequence vector less than or equal to 4 or greater than or equal to 8. What is the command line that you must type to obtain the result ?
7. You want to know if the first element of the sequence vector is less than or equal to 4 or greater than or equal to 8. What is the command line you must type to get the result ?
8. You will now build the truth tables of the logical operators using the `outer` function. The `outer` function allows you to apply a function with two arguments to all the pairs of elements that you can form from the coordinates of two vectors.

Example : You consider the vector `u` of integers from 1 to 10 and the vector `v` of integers from 1 to 8. You want to construct the two-dimensional array with the product of all the coordinates of the two vectors. You run the following command :

```
u<-1:10
v<-1:8
outer(u,v,"*")
```

Run the following commands and notice that they explain the behavior of logical operators if there are missing values.

```
x<-c(NA,FALSE,TRUE)
names(x)<-as.character(x)
!x
outer(x,x,"&")
outer(x,x,"|")
outer(x,x,"xor")
```