Capsule 6 – Les Matrices

Bonjour à tous, je m'appelle Guillaume Dubé et je suis étudiant à la maîtrise en épidémiologie à l'École de santé publique de l'Université de Montréal. Bienvenue dans cette sixième capsule qui abordera le concept des matrices.

Section théorique

Qu'est-ce qu'une matrice ? Une matrice est, selon Wikipédia, un tableau rectangulaire de longueur m par n d'éléments ordonnés en rangées et colonnes qui servent à interpréter les propriétés d'un objet mathématique. Pour les matrices, « m » représente le nombre de rangées (horizontales) et « n » représente le nombre de colonnes (verticales). En d'autres termes, il s'agit d'un vecteur à 2 dimensions. On se souvient qu'un vecteur est une liste de valeur unidimensionnelle (se référer à la capsule 5 au besoin). Telles que les vecteurs, les matrices existent sous 3 formes, soit les matrices numériques, les matrices caractères et les matrices logiques. Il est important de noter que toutes les valeurs incluses dans une matrice seront catégorisées de façon identique. Une colonne ne pourra pas être considérée comme étant numérique alors qu'une autre est caractère. Les matrices peuvent, autant que les vecteurs, être accessibles par des indices.

En résumé, dans le langage de programmation R, un tableau de données à deux dimensions qui contient des valeurs de types identiques se nomme matrice. Nous allons nous familiariser à ce concept dans la section pratique.

Section pratique

Tout d'abord, nous allons créer une matrice 4x3 (4 rangées et 3 colonnes). Il existe plusieurs méthodes pour arriver à nos fins, mais nous allons ici utiliser la fonction matrix().

```
a <- matrix(1:12, nrow = 4, ncol = 3)
```

Lorsque nous appelons l'indice a, nous pouvons voir que R a automatiquement ordonné les valeurs 1 à 12 par colonne. Pour cet exercice, nous allons spécifier à R que nous voulons ordonner nos valeurs par rangée avec l'attribut byrow=TRUE.

```
a <- matrix(1:12, nrow = 4, ncol = 3, byrow = TRUE)
```

L'appel de l'indice « a » nous démontre que la matrice est maintenant ordonnée par rangée. Si nous volons imprimer seulement une colonne précise, par exemple la colonne 3, il est possible de le faire de la façon suivante :

a[,3]

Il est autant possible d'imprimer uniquement la rangée 4.

a[4,]

Il est aussi possible que vous souhaitez imprimer un ensemble précis de colonnes. Pour ce faire, il suffit d'utiliser la fonction suivante. Il est à noter que la même procédure est utilisée pour imprimer un ensemble précis de rangées.

a[,c(1,3)]

ou

a[,-2]

Finalement, il est possible d'imprimer une valeur précise par sa position dans la matrice de la façon suivante :

a[3,1]

Afin de faciliter l'interprétation des matrices, nous pouvons identifier le nom des colonnes et des rangées à l'aide de l'attribut dimnames. Afin d'attribuer des chiffres pour les rangées et des lettres pour les colonnes, il suffit d'utiliser la formule suivante :

dimnames(a) <- list(1:4, letters[1:3])

Tout comme les vecteurs, nous pouvons modifier les valeurs avec la formule suivante :

a[4,2] <- NA

Nous pouvons utiliser une condition pour remplacer toutes les valeurs respectant cette dernière :

a[a<7] <- 4

Il est aussi possible de combiner des matrices. Pour les combiner selon les colonnes, il suffit d'utiliser la fonction cbind(matrice1, matrice2). Pour les combiner selon les rangées, il suffit

d'utiliser la fonction rbind(matrice1, matrice2). Il est à noter que le nombre de rangées doit être identique si nous voulons combiner deux matrices selon leur colonne, et vice-versa. La même règle s'applique aussi pour l'addition, la soustraction, la multiplication et la division de matrices. Pour ce faire, il suffit d'utiliser les signes « + », « - », « * », « / » entre deux matrices.

En mathématique, il s'avère parfois pertinent de transposer des matrices. Pour ce faire, la technique est très simple :

```
b <- t(a)
```

Pour connaître le déterminant de la matrice, il suffit d'utiliser la fonction det(a).

colMeans(a) et rowMeans(a) nous permettent de connaître les moyennes de la matrice selon les colonnes ou les rangées.

Base de données

Ouvrons maintenant nos deux jeux de données et étudions deux variables de la base de données apacheApsVar. Premièrement, à l'aide de la fonction summary, étudions les variables « meanbp » et « heartrate ».

Nous nous apercevons encore une fois qu'il y a des « -1 » que nous allons vouloir modifier pour indiquer qu'il s'agit de données manquantes :

```
apacheApsVar$meanbp[apacheApsVar$meanbp == -1] <- NA
```

```
apacheApsVar$heartrate[apacheApsVar$heartrate == -1] <- NA
```

Afin de combiner deux vecteurs et pour ainsi créer une matrice, nous allons utiliser la fonction cbind.

```
x <- cbind(apacheApsVar$meanbp, apacheApsVar$heartrate)
```

Afin de pouvoir analyser notre matrice, nous allons devoir enlever les données manquantes. Pour ce faire, nous pouvons utiliser la fonction na.omit.

```
x <- na.omit(x)
```

Les données manquantes ont donc été retirées, tel que nous pouvons le vérifier avec la fonction summary(x).

Finalement, l'exercice est de modifier les valeurs de ma matrice selon deux conditions, soit les individus avec une pression sanguine supérieure à 140 ainsi qu'une moyenne de la fréquence cardiaque supérieure à 100. Pour ce faire, nous allons utiliser la fonction ifelse.

```
x[,1] <- ifelse(x[,1] > 140,1,0)

x[,2] <- ifelse(x[,2] > 100,1,0)
```

Afin de calculer un facteur de risque, nous pouvons addition nos deux nouvelles colonnes afin de rapidement identifier les personnes avec une pression élevée ainsi qu'une fréquence cardiaque élevée.

```
z <- x[,1] + x[,2]
```

C'est ainsi que se termine cette 6e capsule. La prochaine capsule traitera des tableaux (dataframe)