Guide pour la simulation de données expérimentales

Sylvain Hanneton

Guide pour la simulations de données expérimentales

I. Introduction

I.a. Le principe

L'objectif de ce document est de vous aider à obtenir des données simulées de façon autonome. Pour cela, il faut utiliser un système de tirage au sort des données. Mais ce tirage au sort ne peut pas se faire n'importe comment. Pour obtenir des données exploitables, il faut en fait choisir le générateur aléatoire qui correspond à la distribution que respecteraient les données expérimentales.

Supposons par exemple que le résultat d'un questionnaire donne un score global situé entre 0 et 30. Et que l'on souhaite obtenir une moyenne de 27. Quelle générateur aléatoire, obéissant à quelle distribution, permettrait d'obtenir le résultat attendu ?

- cela ne peut être un générateur aléatoire avec une distribution uniforme : en effet la moyenne serait de 30 et non 27
- cela ne peut être un générateur aléatoire avec une distribution normale : une distribution normale pourrait être centrée sur 27 mais elle donne des valeurs entre - l'infini et + l'infini.

Donc il faut choisir le générateur aléatoire adapté à l'outil d'évaluation utilisé dans le protocole. Nous allons ainsi passer en revue les solutions possibles pour chaque type d'outil.

Nous allons utiliser le logiciel R pour générer les données.

I.b. Aller chercher des exemples dans la littérature scientifique

Pour obtenir des données simulées les plus proches possibles de celles que vous pourriez obtenir, il est nécessaire d'aller chercher dans la littérature scientifique les résultats obtenus **les plus proches** de ceux que vous auriez pu obtenir :

- en terme de population visée (âge, ratio des sexes etc…)
- dans un protocole le plus proche du votre (nombre de groupes, répétition des mesures etc…)

Vous pouvez alors constituer un tableau résumant les données recueillies dans la littérature en listant pour chaque article, les grandeurs qui vous intéressent (moyennes, écarts-type, probabilité, pourcentages etc…)

Ce tableau servira de base à vos simulations.

II. Comment utiliser les exemples de code ?

C'est très simple : il suffit

- d'installer le logiciel R sur votre ordinateur (ici [https://pbil.univ-lyon1.fr/CRAN/] par exemple).
- de lancer R, qui va affiche une fenêtre de commande
- de copier le code proposé dans le document et le coller dans la fenêtre de commande
- d'appuyer sur ENTREE

Vous pouvez aussi alternativement ouvrir une fenêtre de script dans R, y coller le code, et l'éxécuter à partir de la fenêtre de script.

III. Attention aux mesures répétées !

III.a. Les mesures ne sont pas indépendantes

Il est important de considérer différemment le cas où vous avez deux échantillons de mesures indépendants (deux groupes d'individus par exemple) et le cas où les mesures sont répétées sur les mêmes individus (échantillons non indépendants).

Imaginons par exemple que l'on fasse passer un test à un groupe d'individus avant et après un programme d'activités physiques. On obtient donc pour chaque individu deux mesures. Dans ce cas, la progression d'un individu va dépendre de sa performance initiale :

- un individu qui commence par un bon score, a peu de chance d'obtenir un mauvais score lors de la seconde mesure, par contre sa progression sera éventuellement plus faible
- un individu qui commence par un score médiocre a peu de chance d'avoir un excellent score lors de la seconde mesure, même si sa marge de progression est plus importante.

Les deux échantillons de mesures (avant et après) ne sont pas indépendants : la performance d'un individu dans le second échantillon dépend de sa performance dans le premier échantillon.

III.b. Comment faire ?

Dans ce cas, on ne peut tirer au hasard séparément les deux scores avant et après, car cela reviendrait à avoir deux groupes d'individus différents. On va donc procéder comme suit :

- On tire au hasard les scores initiaux (avant le programme) en choisissant la bonne distribution de probabilité.

- On tire au hasard, pour chaque individu, leur progression attendue, là encore en choisissant la bonne distribution de probabilité.

Les scores finaux des individus seront donc la somme, pour chacun d'eux, du score initial et de la progression.

IV. Variables qualitatives

IV.a. Données dichotomiques

On est en présence de données dichotomiques lorsqu'il n'existe que deux types de réponses possibles. Par exemple VRAI / FAUX, OUI / NON, MASCULIN / FEMININ. Ces données suivent une distribution binomiale.

Exemple 1 : le genre

Supposons que l'on souhaite obtenir dans deux échantillons de 30 personnes une distribution des genre répartie entre 60 % de femmes et 40 % d'hommes. Alors on peut générer cet échantillon en utilisant un générateur aléatoire capable de respecter cette distribution binomiale.

La fonction **rbinom** de R permet de faire de tels tirages. Les arguments à lui donner sont :

- le nombre d'observations, c'est à dire le nombre d'échantillons (ici on veut deux échantillons)
- le nombre de tirages dans chaque observation, c'est à dire ici le nombre de d'individus dans chaque échantillon.
- La probabilité de succès : on considérera ici que le fait d'être une femme est un succés !

```
nech <- 2 ; # nombre d'échantillons souhaités
n <- 30 ; # nombre d'individus dans chaque échantillon
p <- 0.6 ; # probabilité d'obtenir un succès
tirage <- rbinom(nech, size=n, prob=p); # Tirage au sort
print(tirage) ; # affiche le résultat

## [1] 20 19

cat(tirage[1]," femmes dans l'échantillon 1","\n");

## 20  femmes dans l'échantillon 1

cat(tirage[2]," femmes dans l'échantillon 2","\n");

## 19  femmes dans l'échantillon 2
```

Exemple 2 : Réponses OUI / NON dans un questionnaire

Dans un questionnaire, 70 participants doivent répondre oui ou non à une question spécifique. On s'attend à avoir seulemen 10% de réponses oui. Nous n'avons donc ici qu'un seul échantillon.

Dans ce cas on peut obtenir un tirage au sort de la façon suivante :

```
n <- 70 ; # nombre de tirages (d'individus) souhaités
p <- 0.1 ; # probabilité d'obtenir un oui
tirage <- rbinom(1, size=n, prob=p); # Tirage au sort
print(tirage) ; # affiche le résultat
## [1] 7
cat(tirage," réponses OUI dans l'échantillon !","\n");
## 7  réponses OUI dans l'échantillon !
```

Vous noterez que si l'on exécute plusieurs fois la fonction rbinom avec les mêmes paramètres, on obtient un effectif de réponses oui qui peut varier à chaque fois.

IV.b. QCM : Données multinomiales

QCM : Une seule réponse possible ou échelle de Likert

Il s'agit ici de la situation ou pour une question il y a n choix possibles, chaque choix ayant une probabilité différente de se réaliser. Cela revient à jeter un dé à n faces, mais avec des faces qui ont des probabilités différentes d'apparaître. Attention **la somme des probabilités doit faire 1.**

Ce type de tirage obéit à la **loi multinomiale**.

Exemple : A la question "Allez vous voter pour le candidat X ?" la personne peut répondre

- Oui
- Non
- Je ne sais pas

Si l'on suppose que les probabilité de choisir chacune des 3 possibilités sont 40 %, 30 %, 30 %, alors ont peut obtenir le tirage au sort par la fonction **rmultinom**. Pour un seul individu, la commande qui permet d'obtenir un choix aléatoire entre les 3 options est :

```
p <- c(0.4,0.3,0.3) ; # liste des probabilités des différents choix possibles, la somme doit faire 1
tirage <- rmultinom(1, size=1, prob=p) ; # Tirage au sort
print(tirage) ; # affiche le résultat

##      [,1]
## [1,]    0
## [2,]    0
## [3,]    1
```

Pour 10 individus :

```
n <- 10 ; # nombre d'individus (de tirages) souhaités
p <- c(0.4,0.3,0.3) ; # liste des probabilités des différents choix possibles, la somme doit faire 1
tirage <- rmultinom(n, size=1, prob=p) ; # Tirage au sort
print(tirage) ; # affiche le résultat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    0    1    1    1    1    1    0     0
## [2,]    0    0    0    0    0    0    0    0    1     1
## [3,]    0    0    1    0    0    0    0    0    0     0
```

```
print(rowSums(tirage)) ; # affiche la somme des choix pour chaque case
```

```
## [1] 7 2 1
```

Dans le cas d'une échelle de Likert, les probabilités des différents choix peuvent être estimées à partir des distributions de réponses observées dans un travail expérimental.

QCM : plusieurs choix possibles

Ici le participant puisse cocher 2 cases sur 5 choix possibles, sachant que chacun des choix a une probabilité différente de se réaliser. Cela revienbt à tirer au sort deux boules dans une urne qui contient des boules de cinq couleurs différentes sans remise. Et si l'on tire une couleur, alors on ne peut pas retirer une boule de la même couleur.

On peut pour simuler une réponse pour un individu, utiliser la fonction **sample**.

```
nc <- 5 ; # nombre de cases, à remplacer par votre valeur
nchoix <-  2 ; # nombre de cases que l'on a le droit de cocher parmi les nc cases
p <- c(0.2,0.3,0.2,0.1,0.2) ; # probabilité pour chaque case d'être cochée, la somme doit faire 1
tirage <- sample(nc,size=nchoix,replace=FALSE,p) # tirage au sort
print(tirage) ; # affiche le résultat
```

```
## [1] 2 5
```

pour plusieurs individus (ici 10) :

```
n <-10 ; # le nombre d'individus
nc <- 5 ; # nombre de cases, à remplacer par votre valeur
nchoix <-  2 ; # nombre de cases que l'on a le droit de cocher parmi les nc cases
p <- c(0.2,0.3,0.2,0.1,0.2) ; # probabilité pour chaque case d'être cochée, la somme doit faire 1
tirage <- replicate(n,sample(nc,size=nchoix,replace=FALSE,p)) # tirage au sort
print(tirage) ; # affiche le résultat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    3    2    1    2    3    2     3
## [2,]    5    5    2    1    4    5    1    2    5     1
```

V. Variables quantitatives

V.a. Nombres d'évènements se passant dans un temps donné.

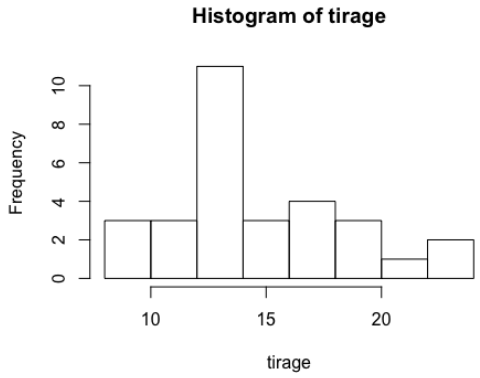
La fonction à utiliser est **rpois** qui simule une distribution de Poisson, avec comme paramètre le nombre d'individus dans l'échantillon ainsi que la moyenne (lambda) du nombre d'évènements qui arrivent dans le temps donné.

Exemple : Test de lever de chaise en 30 secondes. On compte le nombre de levers.Supposons que la moyenne observée soit 16 et que l'échantillon soit composé de 30 personnes alors :

```
n <- 30 ; # nombre de personnes dans l'échantillon
m <- 16 ; # la moyenne attendue, à remplacer par votre valeur
tirage <- rpois(n,m) # tirage au sort
print(tirage) ; # affiche le résultat

##  [1] 19 14 14 13 18 14  9 16 17  9 12 20 13 11 16 13 14 19 16 13 21 23 18
14 13
## [26] 12 18 24 14 10

hist(tirage) ;
```




```
cat("La moyenne est ",mean(tirage)," levers de chaise",". L'écart type est
de :",sd(tirage)) ; # calcule et affiche la moyenne

## La moyenne est  15.23333  levers de chaise . L'écart type est de : 3.838852
```

V.b. Variable continue strictement positive

Une possibilité est d'utiliser un générateur aléatoire suivant la distribution **log-normale**.

En théorie des probabilités et statistique, une variable aléatoire X est dite suivre une loi log-normale si la variable Y = log (X) suit une loi normale.

Ce type de variable peut être un temps chronométré par exemple, ou une valeur comme la VO2Max. La distribution ne sera pas normale car les valeurs **sont bornées à gauche par zéro**. Autrement dit elles sont forcément strictement positives.

On peut utiliser la distribiution **log-normale** avec la fonction **rlnorm()**. Il faut préciser la taille de l'échantillon, la moyenne et l'écart type. Supposons que l'on mesure par exemple le temps de réalisation d'une épreuve. Le temps moyen est 3 minutes (180 secondes) avec un écart-type de 30 secondes.

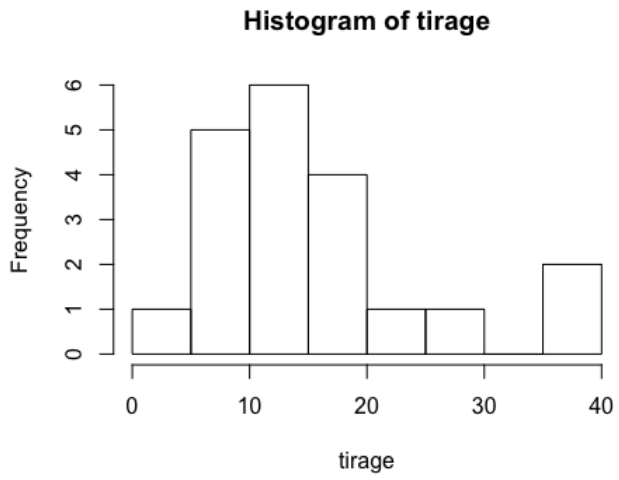
Attention, la moyenne à fournir dans la fonction n'est pas la moyenne dans la variable de mesure. Il faut faire une transformation logarithmique.

Supposons que l'on appelle m et s la moyenne (espérance) et l'écart type des temps mesurés. Alors il faut donner à la fonction :

- comme moyenne (meanlog) la valeur log(m) - 1/2 log(1 + $s^{2/m}2$)
- comme ecart type (sdlog) la valeur racine(log(1 + $s^{2/m}2$))

Le calcul ci-dessous est réalisé pour 1000 individus !

```
m <- 20 ; # la moyenne, a remplacer par la valeur que vous souhaitez
s <- 20.7 ; # l'ecart type, a remplacer par la valeur que vous souhaitez
sl <- sqrt(log(1+ s^2/m^2)) ;
ml <- log(m) - 0.5 * sl^2 ;
tirage <- rlnorm(20,meanlog=ml, sdlog=sl) ; # tirage au sort
hist(tirage) ; # trace l'histogramme des valeurs pour voir la distribution
```




```
cat("La moyenne est ",mean(tirage)," ! ",". L'écart type : ",sd(tirage))  #
calcule et affiche la moyenne
```

```
## La moyenne est  15.90903  !  . L'écart type :  9.932464
```

NB : il est possible d'utiliser également la loi log-logistique ou la loi gamma.

V.c. Variable continue comprise entre deux valeurs

C'est le cas par exemple :

- pour une échelle analogique
- pour un score compris entre 0 et un maximum positif

Il n'est pas du tout évident de trouver une solution pour simuler ce type de données.

Première méthode la plus simple : utiliser la distribution de la loi normale

Le principe de cette méthode est d'utiliser un générateur aléatoire qui suit une loi normale. Mais comme ce type de générateur va fournir des nombres qui seront en principe compris entre moins l'infini et plus l'infini, alors il est probable de tirer au sort des valeurs qui ne seront pas dans l'intervalle souhaité. Tout simplement **on ne prendra pas en compte ces tirages qui ne conviennent pas**.

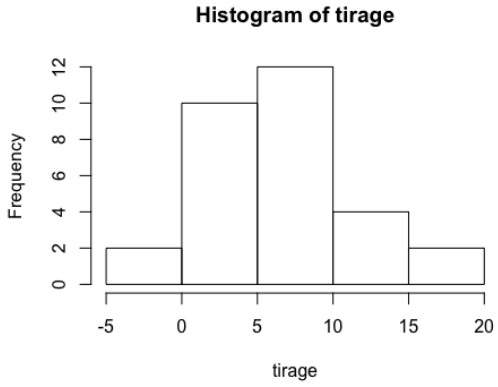
Cette méthode n'est pas rigoureuse mais est la plus simple à mettre en place.

Prenons un exemple : on souhaite obtenir les scores de 30 personnes sachant que ce score doit être compris entre 0 et 10. Dans la littérature on trouve que la moyenne est de 7 et l'écart type de 5.

```
n <-30 ; # le nombre d'individus
m <- 7 ; # la moyenne attendue, à remplacer par votre valeur
s <- 5 ; # l'écart type attendu, à remplacer par votre valeur
tirage <- rnorm(n,mean=m,sd=s) ; # tirage au sort
print(tirage) ; # affiche le résultat
```

```
##  [1] 15.9045831  6.3976355 10.7019782  6.3209960  3.5740128 13.8626735
##  [7]  9.7207693  8.6578045  2.4311692  5.9633581 -0.9146991  1.2008849
## [13] 11.8989595  4.2745975  9.1972890 -0.5225601  0.1527590  0.4651509
## [19]  4.7561508 13.8206521  2.4757802  6.7726461  7.6521361  6.8917587
## [25]  3.5610540 15.2437662  6.6403773  2.6055996  6.7218516  9.4287950
```

```
hist(tirage) ; # distribution (histogramme) des résultats
```



```
cat("La moyenne et l'écart type obtenus
sont :",mean(tirage),"et",sd(tirage),"\n") ;
```

```
## La moyenne et l'écart type obtenus sont : 6.528598 et 4.677564
```

```
cat("Les valeurs minimales et maximales sont respectivement ",min(tirage), "
et ",max(tirage)) ;
```

```
## Les valeurs minimales et maximales sont respectivement  -0.9146991  et
15.90458
```

Dans les résultats obtenus ci-dessus, vous allez forcément avoir de grande chance d'obtenir des valeurs inférieures à 0 ou supérieures à 10. Il ne faudra donc pas retenir ces valeurs et refaire d'autres tirages pour parvenir au nombre souhaité pour votre échantillon.

Seconde méthode : utiliser la loi binomiale

A suivre…. (en cours de rédaction)
