Exercices_2_solution

May 22, 2018

1 Question 1

Créer deux vecteurs aléatoires nommés `x1` et `x2`, contenant chacun 100 observations pseudo-aléatoire d'une distribution:

(1)

1. de loi normale centrée réduite et
2. de loi uniforme définie sur l'intervalle [0 ; 10].

```
In [2]: set.seed(123)
        x1<-rnorm(100)
        x2<-runif(100, 0,10)
```

Créez une matrice 10X10 contient les valeur du vecteur x1 crée auparavant:

```
In [3]: matNorm<-matrix(x1,nrow=10)
```

Calculez la moyenne de cette dernière et la variance de cette dernière

```
In [4]: mean(matNorm)
```

0.0904059086362066

```
In [5]: (sd(matNorm))**2
```

0.833232830197759

2 Question 2

Créer un vecteur xx1 contenant un échantillon équiprobable de 4 variables à partir du vecteur x1 de la question précédente

```
In [6]: xx1<-sample(x1, 4)
```

À partir du vecteur xx1, créez un autre vecteur xx2 qui possède 1000 variables de l'échantillon xx1. La dernière variable possède une probabilité de 70% qu'elle soit tirée alors que les trois premières ont chacune 10% de chance qu'elle soit tirée.

```
In [7]: xx2<-sample(xx1, 1000, replace = T, prob=c(.1,.1,.1,.7))
```

Donnez la fréquence de chacune des variables simulée

```
In [8]: table(xx2)
```

```
xx2
-0.560475646552213 -0.491031166056535 -0.284773007051009   0.18130347974915
               113                702                100                  85
```

3 Question 3

On vous dit que les temps pour finir un demi-triathlon suivent une loi normale avec une moyenne (et les écarts types ET) pour les hommes et les femmes sont les suivantes:

- Pour les **hommes** nager 1.9 km en 40 minutes (ET=3), pédaler 90 km en 2:45 (ET=8), et courir 21.1 km en 2:05 (ET=10).
- Pour les **Femmes** nager 1.9 km en 50 minutes (ET=5), pédaler 90 km en 3:00 (ET=5), et courir 21.1 km en 2:15 (ET=12).

Créez les vecteurs {`swimH`, `bikeH`, `runH`, `swimF`, `bikeF`, `runF`} contenant le temps pour chacun des sports pour 1002 hommes et 1300 femmes, tirés aléatoirement selon les lois ci-dessus (on suppose que les trois sports sont indépendants même si en réalité ce n'est jamais vrai, car si on se blesse en vélo, on performe beaucoup moins en course).

```
In [9]: set.seed(123)
        n<-1002
        swimH<-round(rnorm(n, mean = 40, sd = 3),2)
        bikeH<-round(rnorm(n, mean = 165, sd = 8),2)
        runH<-round(rnorm(n, mean = 125, sd = 8),2)

        m<-1300
        swimF<-round(rnorm(m, mean = 50, sd = 5),2)
        bikeF<-round(rnorm(m, mean = 180, sd = 5),2)
        runF<-round(rnorm(m, mean = 135, sd = 12),2)
```

Avec les vecteurs crée précedemment, construisez une matrice pour les hommes et une autre pour les femmes

```
In [10]: resultatH<-matrix(c(swimH, bikeH, runH), ncol = 3)
         colnames(resultatH)<-c("Swim", "Bike", "Run")
         rownames(resultatH)<-paste("H",1:n,sep='')
```

```
In [11]: head(resultatH)
```

	Swim	Bike	Run
H1	38.32	164.86	126.39
H2	39.31	163.94	120.08
H3	44.68	144.61	110.54
H4	40.21	173.32	119.85
H5	40.39	167.00	141.37
H6	45.15	184.33	120.51

```
In [12]: tail(resultatH)
```

	Swim	Bike	Run
H997	43.21	169.29	123.80
H998	35.95	161.32	122.38
H999	38.43	160.91	113.41
H1000	39.25	166.90	119.42
H1001	37.01	160.67	145.79
H1002	36.88	174.75	124.70

```
In [13]: resultatF<-matrix(c(swimF, bikeF, runF), ncol = 3)
         colnames(resultatF)<-c("Swim", "Bike", "Run")
         rownames(resultatF)<-paste("F",n+1:m,sep='')
```

```
In [14]: head(resultatF)
```

	Swim	Bike	Run
F1003	54.57	180.26	117.59
F1004	49.08	184.04	130.86
F1005	53.05	175.30	115.72
F1006	49.74	180.20	149.05
F1007	56.82	170.01	131.69
F1008	47.48	180.69	132.66

```
In [15]: tail(resultatF)
```

	Swim	Bike	Run
F2297	57.17	172.19	114.79
F2298	54.56	174.28	132.75
F2299	51.91	176.39	132.54
F2300	52.76	182.63	126.28
F2301	50.72	168.28	124.29
F2302	58.54	179.21	134.37

Créez une **seule** matrice qui contient les résultats des femmes ensuite et le résultat des hommes

```
In [16]: résultat<-rbind(resultatH, resultatF)
```

```
In [17]: head(résultat)
```

	Swim	Bike	Run
H1	38.32	164.86	126.39
H2	39.31	163.94	120.08
H3	44.68	144.61	110.54
H4	40.21	173.32	119.85
H5	40.39	167.00	141.37
H6	45.15	184.33	120.51

```
In [18]: tail(résultat)
```

	Swim	Bike	Run
F2297	57.17	172.19	114.79
F2298	54.56	174.28	132.75
F2299	51.91	176.39	132.54
F2300	52.76	182.63	126.28
F2301	50.72	168.28	124.29
F2302	58.54	179.21	134.37

Quel est le numéro du dossard du participant/es qui a le meilleur temps en nage, et en combien de temps à accomplie cette discipline

```
In [19]: min(résultat[, 1])
```

31.57

```
In [20]: which(résultat[, 1]==min(résultat[, 1]))
```

H591: 591

Quel est le numéro du dossard du participant/es qui a le meilleur temps en vélo, et en combien de temps à accomplie cette discipline

```
In [21]: which(résultat[, 2]==min(résultat[, 2]))
```

H701: 701

```
In [22]: min(résultat[, 2])
```

140.62

Quel est le numéro du dossard du participant/es qui a le meilleur temps en course, et en combien de temps à accomplie cette discipline

```
In [23]: which(résultat[, 3]==min(résultat[, 3]))
```

F1559: 1559

```
In [24]: min(résultat[, 3])
```

95.53

Quel a été le meilleur temps chez les femmes?

```
In [25]: tempsF<-resultatF[, 1]+resultatF[, 2]+resultatF[, 3]
```

```
In [26]: bestF<-min(tempsF)
         bestF
```

321.58

Quel numéro de dossard?

```
In [27]: which(tempsF==min(tempsF))
```

F1666: 664

Qui a gagné la course et en combien de temps?

```
In [28]: temps<-résultat[, 1]+résultat[, 2]+résultat[, 3]
         gagnant<-min(temps)
         gagnant
         which(temps==min(temps))
```

296.08
H614: 614

4 Question 4

Créez un vecteur appelé *ann* de qui représente les années de développement dans calcul d'annuité de 5 ans, qui donne le résultat suivant $\{1,\ldots,5\}$

```
In [29]: ann<-1:5
         ann
```

1. 1 2. 2 3. 3 4. 4 5. 5

Créez un vecteur contenant les fameux facteurs d'actualisation v^n qui servent à calculer la valeur présente d'une série de paiements $n = 5$ avec un taux d'intérêt de 2.5%

$$v^n = \frac{1}{1+i} \tag{2}$$

```
In [30]: i<-.025
         v_n<-(1+i)**(-ann)
         v_n
```

1. 0.975609756097561 2. 0.951814396192743 3. 0.928599410919749 4. 0.905950644799755 5. 0.883854287609517

Calculez la valeur présente d'une annuité 5 ans avec qui 153.25$ par année

$$\begin{aligned} PV &= a_n \\ &= v + v^2 +, \ldots, v^n \\ &= \sum_{j=1}^{n} v^j \end{aligned} \tag{3}$$

```
In [31]: pmt<-153.25
         sum(pmt*v_n)
```

711.973216953662

Reproduisez votre calcul avec la fonction suivante:

$$\begin{aligned} PV &= a_n \\ &= v + v^2 +, \ldots, v^n \\ &= \sum_{j=1}^{n} v^j \\ &= \frac{1 - v^n}{i} \end{aligned} \tag{4}$$

Lorsque le taux d'intérêt est constant d'une année à l'autre

```
In [32]: n<-5
         i<-.025
```

```
In [33]: vn<-(1+i)**-n
         vn
```

0.883854287609517

```
In [34]: PV<-(1-vn)/i
         PV
```

4.64582849561931

```
In [35]: pmt*PV
```

711.973216953659

5 Question 5

on se rappelle du taux *Effective rate of discount*

$$d_t = \frac{a(t) - a(t-1)}{a(t-1)} \tag{5}$$

Le taux *discount* se calcule avec la fonction suivante:

$$d = \frac{i}{1+i} = iv \tag{6}$$

Soit un taux d'intérêt de 5%, quel sera alors de taux de *discount* avec seulement 6 décimales

```
In [36]: i<-.05
         d<-i/(1+i)
         round(d,6)
```

0.047619

6 Question 6

Écrivez un code R pour créer la liste suivante :

```
In [37]: (x <- list(ssd = c(256, 128, 512), machine = "Macbook Pro", best = TRUE))
```

$ssd 1. 256 2. 128 3. 512

$machine 'Macbook Pro'

$best TRUE

Ecrivez un code qui extrait les différente tailles du ssd **seulement**

```
In [38]: x[[1]]
```

1. 256 2. 128 3. 512

```
In [39]: x$ssd
```

1. 256 2. 128 3. 512

Extraire les étiquettes de la liste;

```
In [40]: names(x)
```

1. 'ssd' 2. 'machine' 3. 'best'

Extraire le 3e élément du premier élément de liste:

```
In [41]: x[[1]][3]
```

512

Remplacer le dernier élément par le vecteur `T,F,T`

```
In [42]: x[[3]]<-c(T,F,T)
```

```
In [43]: x
```

$ssd 1. 256 2. 128 3. 512

$machine 'Macbook Pro'

$best 1. TRUE 2. FALSE 3. TRUE

7 Question 7

Soit le vecteur suivant:

```
In [44]: x<-c(71,18,86,5,58,19,14,9,74,75,59,24,7,51,50,63,35,53,72,61)
         x
```

1. 71 2. 18 3. 86 4. 5 5. 58 6. 19 7. 14 8. 9 9. 74 10. 75 11. 59 12. 24 13. 7 14. 51 15. 50 16. 63 17. 35 18. 53 19. 72 20. 61

Extraire le 10e élément du vecteur;

```
In [45]: x[10]
```

75

Extraire une partie du vecteur allant composé du 1er, 3e, ..., 19e élément

```
In [46]: x[seq(from = 1, to = 19, by = 2)]
```

1. 71 2. 86 3. 58 4. 14 5. 74 6. 59 7. 7 8. 50 9. 35 10. 72

Extraire les éléments divisibles par deux (*even numbers*)

```
In [47]: x[x%%2==0]
```

1. 18 2. 86 3. 58 4. 14 5. 74 6. 24 7. 50 8. 72

Extraire les éléments non divisibles par deux (*odd numbers*)

```
In [48]: x[x%%2!=0]
```

1. 71 2. 5 3. 19 4. 9 5. 75 6. 59 7. 7 8. 51 9. 63 10. 35 11. 53 12. 61

Tous les éléments sauf 3e, 5e et 17e éléments

```
In [49]: x[-c(3, 5, 17)]
```

1. 71 2. 18 3. 5 4. 19 5. 14 6. 9 7. 74 8. 75 9. 59 10. 24 11. 7 12. 51 13. 50 14. 63 15. 53 16. 72 17. 61

Dans le vecteur `x`, combien d'éléments sont pairs et combien sont impairs

```
In [50]: length(x[x%%2==0])
```

8

```
In [51]: length(x[x%%2!=0])
```

12

8 Question 8

Soit une matrice 12X7,

```
In [52]: x <- matrix(sample(1:100, 12*7), 12, 7)
         x
```

97	55	61	20	86	87	35
69	52	75	9	72	57	43
36	23	30	14	10	78	59
83	100	85	24	53	40	1
84	4	6	33	66	91	71
79	63	92	21	62	70	60
38	25	3	8	74	81	5
26	46	64	49	39	17	65
58	68	11	47	54	45	50
95	82	96	29	16	88	37
89	93	34	15	90	32	41
2	94	27	7	76	51	12

---

extraire l'élément de la 5e ligne et 6e colonne

```
In [53]: x[5,6]
```

91

Extraire **tout** le contenu de la 3e ligne et la 9 ligne

```
In [54]: x[c(3,9),]
```

36	23	30	14	10	78	59
58	68	11	47	54	45	50

Extraire **tout** le contenu des colonnes impaires

```
In [55]: x[,c(seq(from = 1, to = 7, by = 2))]
```

97	61	86	35
69	75	72	43
36	30	10	59
83	85	53	1
84	6	66	71
79	92	62	60
38	3	74	5
26	64	39	65
58	11	54	50
95	96	16	37
89	34	90	41
2	27	76	12

9 Questions 9 mathématiques financière

$$1+i=\left(1+\frac{i^{(m)}}{m}\right)^{m}=(1-d)^{-1}=\left(1-\frac{d^{(m)}}{m}\right)^{-m}=e^{\delta} \tag{7}$$

9.1 Q1

En tenant compte de l'équation (7) Quelle est la valeur présente (arrondi à deux décimales) de 1000$ que vous aller recevoir dans 6 et 1/4 avec un taux *effective rate of discount* de 9.27% par année

```
In [56]: PV<-round(1000*(1-.0927)^6.25,2)
         PV
```

544.43

9.2 Q2

En tenant compte de l'équation (7) Quelle est la valeur accumulée (arrondi à deux décimales) de 1300$ que vous aller recevoir dans 10 et 1/2 avec un taux *effective rate of discount* de 5.3286% par année

```
In [57]: PV<-round(1300*(1-.053286)^-10.5,2)
         PV
```

2310.18

9.3 Q3

En tenant compte de l'équation (7) Quelle est la valeur accumulée (arrondi à deux décimales) de 50232$ que vous aller recevoir dans 17 ans avec un taux *nominal rate of interest* de 13% par année convertible trimestriellement?

```
In [58]: FV=round(50232*(1+.13/4)^(17*4),2)
         FV
```

442083.77

9.4 Q4

En tenant compte de l'équation (7) Calculer la valeur présente de 82309$ à payer dans 8 ans avec un taux *nominal rate of discount* de 6% par année composée mensuellement

```
In [59]: FV<-round(82309*(1-(.06/12))^(12*8),2)
         FV
```

50870.16

9.5 Q5

Calculez la valeur présente d'une *annuity-immediate* avec des paiements de 50$ chaque 6 mois pour 10 ans au taux d'intérêt nominal de 4% composé semi-annuellement:

```
In [60]: pmts<-rep(50, 20)
         pmts
```

1. 50 2. 50 3. 50 4. 50 5. 50 6. 50 7. 50 8. 50 9. 50 10. 50 11. 50 12. 50 13. 50 14. 50 15. 50 16. 50 17. 50 18. 50 19. 50 20. 50

```
In [61]: actu<-1:20
         actu
```

1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11 12. 12 13. 13 14. 14 15. 15 16. 16 17. 17 18. 18 19. 19 20. 20

```
In [62]: sum(pmts*(1.02)^-actu)
```

817.571667229856

10 Question 10 probabilité

10.1 Q10.1

Simulez des valeurs tirées d'une distribution normale. Imaginez une population dont la taille moyenne est de 1.70m et un écart-type de 0.1m. En utilisant `rnorm` simulez 100 valeurs et sauvegardez cer dernières dans un vecteur appelé `taille`.

Note Fixez votre seed à une valeur **123**

```
In [63]: set.seed(123)
         taille <- rnorm(n = 100, mean = 1.70, sd = .1)
```

Donnez un sommaire des statistiques descriptives du vecteur `taille`

```
In [64]: summary(taille)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.469   1.651   1.706   1.709   1.769   1.919
```

10.2 Q10.2

Quelle est la probabilité qu'une personne soir plus petit que 1.90m ? Votre réponse arrondie à deux décimales

```
In [65]: round(pnorm(1.90, mean = 1.70, sd = .1),2)
```

0.98

Note

Si on veut les formats en pourcentage, on peut utiliser la fonction `percent` du package `formattable`

```
install.packages("formattable")
```

```
In [66]: library(formattable)
```

```
In [67]: percent(pnorm(1.90, mean = 1.70, sd = .1))
```

97.72%

Quelle est la probabilité que la taille d'une personne soit plus grande que 1.60 m

```
In [68]: percent(1-pnorm(1.60, mean = 1.70, sd = .1))
```

84.13%

10.3 Q10.3

Le temps d'attente (en minute) dans une clinique suit une loi exponentielle avec un taux de 1/50. Utiliser la fonction `rexp` afin de simuler les tempes d'attente pour 30 personnes dans cette clinique.

```
In [69]: set.seed(123)
         (patients <- rexp(rate = 1/50, n =30))
```

1. 42.1728630529201 2. 28.8305135443807 3. 66.4527433903372 4. 1.5788679554156 5. 2.81054880470037 6. 15.8250608188855 7. 15.7113646107649 8. 7.26334019564092 9. 136.311823216485 10. 1.45767235412863 11. 50.2415028845376 12. 24.0107363829887 13. 14.0506813768297 14. 18.8558915533567 15. 9.41420204471797 16. 42.4893064869052 17. 78.1601769807649 18. 23.9380208168278 19. 29.5467417687178 20. 202.050585568625 21. 42.1574865566887 22. 48.2935605549669 23. 74.2637897009036 24. 67.4022242871529 25. 58.4264492129392 26. 80.2926171529007 27. 74.8371434355986 28. 78.5326273447613 29. 1.5883871980738 30. 29.8924845643342

Quelle est la probabilité qu'une personne attende moins que 10 minutes?

```
In [70]: percent(pexp(q = 10,rate = 1/50))
```

18.13%

Supposons que la patience des gens atteint sa limite au bout de 60 minutes. Ça veut dire que s'ils attendent plus que 60 minutes, ils quittent la salle.

S'il y'a 100 personnes dans la salle, combien vont-ils quitter la salle?

```
In [71]: percent(1 - pexp(q=60, rate =1/50))
```

30.12%