

2_4_Solutions

October 15, 2018

Liste des numéros

- 1 Question
- 2 Question
- 3 Question
- 3.1 autre façon de faire
- 4 Question
- 5 Question
- 6 Question
- 7 Question
- 8 Question
- 9 Question
- 10 Question
- 11 Question
- 12 Question
- 13 Questions mathématiques financière
- 13.1 Q1
- 13.2 Q2
- 13.3 Q3
- 13.4 Q4
- 13.5 Q4
- 14 Question probabilité
- 14.1 Q1
- 14.2 Q2
- 14.3 Q3

1 Question

Créer une matrice A au format 2X4 avec les valeurs suivantes: 1 2 3 4 5 6 7 8

```
In [93]: 1:8
```

```
1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8
```

```
In [101]: A<- matrix(c(1:8), nrow = 2, ncol = 4)
```

A

```
1 3 5 7
2 4 6 8
```

Recréez cette matrice, mais cette fois les valeurs sont incrémentées par lignes

```
In [103]: B<- matrix(c(1:8), nrow = 2, ncol = 4, byrow = T)
          B
```

```
 1  2  3  4
 5  6  7  8
```

2 Question

Créez une matrice carrée avec les valeurs du vecteur vec3 créée auparavant:

```
In [108]: matrix(vec3, nrow = 5)
```

```
 2  12  22  32  42
 4   14  24  34  44
 6   16  26  36  46
 8   18  28  38  48
10   20  30  40  50
```

3 Question

Créer une matrice identité 7X7

```
In [30]: mat<-diag(7)
          mat
```

```
 1  0  0  0  0  0  0
 0  1  0  0  0  0  0
 0  0  1  0  0  0  0
 0  0  0  1  0  0  0
 0  0  0  0  1  0  0
 0  0  0  0  0  1  0
 0  0  0  0  0  0  1
```

Afin des fins de calcul de réserves actuarielles, vous devez renverser (*reverse*) la diagonale de sorte où les 1 sont dans les cases (7,1), (6,2),..., (1,7)

```
In [31]: apply(diag(7), 2, rev)
```

```
 0  0  0  0  0  0  1
 0  0  0  0  0  1  0
 0  0  0  0  1  0  0
 0  0  0  1  0  0  0
 0  0  1  0  0  0  0
 0  1  0  0  0  0  0
 1  0  0  0  0  0  0
```

3.1 autre façon de faire

```
In [32]: mat <- mat[ nrow(mat):1, ]
      mat
      0 0 0 0 0 0 1
      0 0 0 0 0 1 0
      0 0 0 0 1 0 0
      0 0 0 1 0 0 0
      0 0 1 0 0 0 0
      0 1 0 0 0 0 0
      1 0 0 0 0 0 0
```

4 Question

À partir des données [suivantes](#), créez une matrice où vous avez la population des villes (Montréal, Québec, Laval, Gatineau) par ligne et les années 2013 à 2016 par colonnes

```
In [33]: pop<-matrix(c(1718241,1735096,1746940,1767753,
      530474,533857,536013,538918,
      417325,421959,425481,429413,
      274180,276290,278050,281392), ncol=4, byrow = T)
```

```
In [34]: pop
      1718241  1735096  1746940  1767753
      530474   533857   536013   538918
      417325   421959   425481   429413
      274180   276290   278050   281392
```

```
In [35]: rownames(pop)<-c("Montréal", "Québec", "Laval", "Gatineau")
      colnames(pop)<-2013:2016
```

```
In [36]: pop
      2013      2014      2015      2016
Montréal 1718241 1735096 1746940 1767753
Québec   530474  533857  536013  538918
Laval    417325  421959  425481  429413
Gatineau 274180  276290  278050  281392
```

On vous dit que l'arrondissement Hochelaga-Maisonneuve (situé à Montréal) est devenu un quartier très aisé et veut maintenant avoir son indépendance. Aujourd'hui ce prestigieux quartier appelé HOMA, l'évolution de la population de ce quartier de 2013 à 2016 a été la suivante: 20000, 20500, 23000, 23800

Quelle aurait été la population de Montréal sans compter les habitants du pays très prospère pays HOMA

```
In [37]: vec_HOMA<-c(20000, 20500, 23000, 23800)
      HOMA<-matrix(vec_HOMA, ncol=4, )
      rownames(HOMA)<-c("HOMA")
      colnames(HOMA)<-2013:2016
      HOMA
```

	2013	2014	2015	2016
HOMA	20000	20500	23000	23800

In [38]: `pop[1,]-vec_HOMA`

2013 1698241 **2014** 1714596 **2015** 1723940 **2016** 1743953

Reconstruisez la nouvelle matrice avec les nouvelles données de Montréal.

In [39]: `pop[1,]<-pop[1,]-vec_HOMA`
`pop`

	2013	2014	2015	2016
Montréal	1698241	1714596	1723940	1743953
Québec	530474	533857	536013	538918
Laval	417325	421959	425481	429413
Gatineau	274180	276290	278050	281392

Ajouter les données de HOMA à la matrice modifiée l

In [173]: `rbind(pop,HOMA)`

	2013	2014	2015	2016
Montréal	1698241	1714596	1723940	1743953
Québec	530474	533857	536013	538918
Laval	417325	421959	425481	429413
Gatineau	274180	276290	278050	281392
HOMA	20000	20500	23000	23800

5 Question

Créer deux vecteurs aléatoires nommés « x1 » et « x2 », contenant chacun 100 valeurs aléatoires compatibles 1. avec une distribution de loi normale centrée réduite et 2. avec une distribution de loi uniforme définie sur l'intervalle [0 ; 10].

In [42]: `x1<-rnorm(100)`
`x2<-runif(100, 0,10)`

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

In [43]: `matNorm<-matrix(x1,nrow=10)`

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

In [44]: `mean(matNorm)`

0.0178254903830712

In [45]: `(sd(matNorm))**2`

0.823977976790721

6 Question

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

```
In [46]: xx1<-sample(x1, 4)
         xx1
```

```
1. -0.163930968642975 2. 0.300279118120242 3. 0.596425059015276 4. -0.488922835294287
```

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

Donnez la fréquence de chacune des variables simulée

```
In [48]: table(xx2)
```

```
xx2
-0.488922835294287  -0.163930968642975  0.300279118120242  0.596425059015276
              705              82              111              102
```

7 Question

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).

Créer 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 [49]: 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 créés précédemment, construisez une matrice pour les hommes et une autre pour les femmes

```
In [50]: resultatH<-matrix(c(swimH, bikeH, runH), ncol = 3)
         colnames(resultatH)<-c("Swim", "Bike", "Run")
         rownames(resultatH)<-paste("H",1:n,sep='')
```

```
In [51]: 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 [52]: 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 [53]: resultatF<-matrix(c(swimF, bikeF, runF), ncol = 3)
         colnames(resultatF)<-c("Swim", "Bike", "Run")
         rownames(resultatF)<-paste("F",n+1:m,sep='')
```

```
In [54]: 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 [55]: 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 matrice qui contient les résultats des femmes ensuite et le résultat des hommes

```
In [292]: resultat<-rbind(resultatH, resultatF)
```

```
In [293]: head(résultat)
```

	Swim	Bike	Run
H1	40.54	171.39	135.25
H2	41.78	157.40	130.89
H3	46.27	168.04	131.96
H4	39.01	157.34	133.89
H5	36.20	160.56	118.65
H6	42.51	163.69	123.67

```
In [294]: tail(résultat)
```

	Swim	Bike	Run
F2297	57.86	176.24	173.19
F2298	48.88	177.83	111.61
F2299	43.47	175.25	137.77
F2300	46.19	180.49	156.81
F2301	51.36	183.77	129.70
F2302	59.24	182.38	134.04

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 [273]: min(résultat[, 1])
```

31.34

```
In [272]: which(résultat[, 1]==min(résultat[, 1]))
```

H19: 19

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

H759: 759

```
In [275]: min(résultat[, 2])
```

142.31

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

F1719: 1719

```
In [277]: min(résultat[, 3])
```

102.02

Quel a été le meilleur temps chez les femmes?

```
In [298]: tempsF<-resultatF[, 1]+resultatF[, 2]+resultatF[, 3]
```

```
In [301]: bestF<-min(tempsF)
          bestF
```

322.77

Quel numéro de dossard?

```
In [302]: which(tempsF==min(tempsF))
```

F1871: 869

Qui a gagné la course et en combien de temps?

```
In [306]: temps<-résultat[, 1]+résultat[, 2]+résultat[, 3]
          gagnant<-min(temps)
          gagnant
          which(temps==min(temps))
```

292.15

H249: 249

8 Question

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

```
In [42]: ann<-1:5
          ann
```

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

Créer 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%

$$v^n = \frac{1}{1+i} \quad (1)$$

```
In [44]: i<-.025
          v_n<-(1+i)**(-ann)
          v_n
```

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

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

$$\begin{aligned} PV &= a_n \\ &= v + v^2 + \dots + v^n \\ &= \sum_{j=1}^n v^j \end{aligned} \quad (2)$$


```
In [46]: pmt<-153.25
         sum(pmt*v_n)
```

711.973216953662

Reproduire votre calcul avec la fonction suivante:

$$\begin{aligned}
 PV &= a_n \\
 &= v + v^2 + \dots + v^n \\
 &= \sum_{j=1}^n v^j \\
 &= \frac{1 - v^n}{i}
 \end{aligned} \tag{3}$$

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

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

```
In [60]: vn<-(1+i)**-n
        vn
```

0.883854287609517

```
In [63]: PV<-(1-vn)/i
        PV
```

4.64582849561931

```
In [64]: pmt*PV
```

711.973216953659

9 Question

on se rappelle du taux *Effective rate of discount*

$$d_t = \frac{a(t) - a(t-1)}{a(t-1)} \tag{4}$$

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

$$d = \frac{i}{1+i} = iv \tag{5}$$

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

```
In [70]: i<- .05
        d<-i/(1+i)
        round(d,6)
```

0.047619

10 Question

Écrivez un code R pour créer la liste suivante :

```
In [349]: (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 différentes tailles du ssd **seulement**

```
In [350]: x[[1]]
```

```
1. 256 2. 128 3. 512
```

```
In [351]: x$ssd
```

```
1. 256 2. 128 3. 512
```

Extraire les étiquettes de la liste;

```
In [352]: names(x)
```

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

Extraire le 3e élément du premier élément de liste:

```
In [353]: x[[1]][3]
```

```
512
```

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

```
In [354]: x[[3]]<-c(T,F,T)
```

```
In [355]: x
```

```
$ssd 1. 256 2. 128 3. 512
```

```
$machine 'Macbook Pro'
```

```
$best 1. TRUE 2. FALSE 3. TRUE
```

11 Question

Soit le vecteur suivant:

```
In [362]: 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 [363]: x[10]
```

75

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

```
In [367]: 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 [371]: 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 [372]: 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 [375]: 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 [373]: length(x[x%%2==0])
```

8

```
In [374]: length(x[x%%2!=0])
```

12

12 Question

Soit une matrice 12X7,

```
In [56]: 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 [379]: x[5,6]

45

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

In [380]: x[c(3,9),]

96	87	17	77	86	1	69
72	3	97	19	27	10	56

Extraire **tout** le contenu des colonnes impaires

In [385]: x[,c(seq(from = 1, to = 7, by = 2))]

40	95	100	16
47	89	62	43
96	17	86	69
81	23	55	25
63	68	88	2
59	51	46	58
31	80	83	39
78	26	32	37
72	97	27	56
98	60	90	64
30	41	73	49
76	48	14	13

13 Questions mathématiques financière

$$1 + i = \left(1 + \frac{i^{(m)}}{m}\right)^m = (1 - d)^{-1} = \left(1 - \frac{d^{(m)}}{m}\right)^{-m} = e^{\delta} \quad (6)$$

13.1 Q1

En tenant compte de l'équation (6) 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 [393]: PV<-round(1000*(1-.0927)^6.25,2)
          PV
```

544.43

13.2 Q2

En tenant compte de l'équation (6) 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 [22]: PV<-round(1300*(1-.053286)^-10.5,2)
          PV
```

2310.18

13.3 Q3

En tenant compte de l'équation (6) 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 [396]: FV=round(50232*(1+.13/4)^(17*4),2)
          FV
```

442083.77

13.4 Q4

En tenant compte de l'équation (6) 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 [397]: FV<-round(82309*(1-(.06/12))^(12*8),2)
          FV
```

50870.16

13.5 Q4

Calculer 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 [399]: 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 [401]: 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 [406]: sum(pmts*(1.02)^-actu)

817.571667229856
```

14 Question probabilité

14.1 Q1

Simuler une 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 ces dernières dans un objet de type vecteur appelé `taille`.

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

```
In [12]: set.seed(123)
          taille <- rnorm(n = 100, mean = 1.70, sd = .1)
```

Donnez un sommaire des statistiques descriptives du vecteur `taille`

```
In [13]: summary(taille)
```

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

14.2 Q2

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

```
In [14]: 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`

```
In [7]: install.packages("formattable")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' ... done
```

```
In [8]: library(formattable)
```

```
In [15]: 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 [10]: percent(1-pnorm(1.60, mean = 1.70, sd = .1))
```

84.13%

14.3 Q3

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 temps d'attente pour 30 personnes dans cette clinique.

```
In [16]: 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 [18]: 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 [21]: percent(1 - pexp(q=60, rate =1/50))
```

30.12%