SY01 PROJET - R

Probabilité et programmation

Printemps 2024

BUTIN Louis

UTT
UNIVERSITÉ DE TECHNOLOGIE
TROYES

Introduction :

Ce projet est un projet de Probabilité en R, il constitue en l'analyse d'une réaction en chaîne dans le domaine nucléaire simulée par un processus de Galton-Watson, les différentes questions des exercices sont disponibles en PDF afin de mieux comprendre les questions.

Ce projet à pour but de s'entraîner à utiliser le langage R et de répondre à des problématiques mathématiques à l'aide de ce dernier.

Ce rapport est constitué des résultats trouvés grâce à R et à l'analyse de ces derniers afin de répondre aux questions posées, le Code R est lui aussi entièrement disponible si besoin.

Exercice 1

Question 1)

Dans cette question, nous souhaitons créer une fonction qui simule l'état de la population à la génération n.

Pour cela, notre fonction implémente le processus de Galton-Watson avec les caractéristiques suivantes :

1. Pour n = 0, on commence avec 1 individu
2. Pour chaque génération n > 0 :
    - On récupère d'abord l'état de la population à la génération précédente
    - Pour chaque individu de la génération précédente, on simule le nombre d'enfants selon une loi de Bernoulli(1/2)
    - On somme le nombre total d'enfants pour obtenir la nouvelle génération

Il est important de prendre en compte le fait que : $P\left[\frac{x_{i,n}}{2} = 1\right] = \frac{1}{2}$ donc : $P[x_{i,n} = 2] = \frac{1}{2}$, ainsi chaque individu peut avoir 0 ou 2 enfants.

J'ai choisi dans cette fonction, afin de vérifier que tout fonctionne bien d'afficher le nombre d'individus à chaque tour de ma boucle, j'ai choisi dans cette fonction, afin de vérifier que tout fonctionne bien d'afficher le nombre d'individus à chaque tour de ma boucle, ainsi je peux voir le nombre d'individus pour chaque génération jusqu'au n donné et vérifier qu'une fois qu'une génération arrive à 0 individus, les autres d'après le sont, j'obtiens les résultats suivants :

```
Génération 0 : 1
Génération 1 : 2
Génération 2 : 2
Génération 3 : 2
Génération 4 : 4
Génération 5 : 2
```

```
Génération 0 : 1
Génération 1 : 2
Génération 2 : 2
Génération 3 : 0
Génération 0 : taille moyenne = 0.978
```

Ainsi, pour la première simulation j'obtiens des résultats cohérents et possibles, il y a bien 1 individu pour la première génération la simulation va jusqu'à la 5e génération comme demandé. Pour la deuxième simulation pour la deuxième simulation, la simulation s'arrête à la génération 3 car on tombe à 0 et que j'ai demandé à ma fonction de ne plus générer de génération s'il y a extinction (si individu = 0).

Donc la fonction fonctionne correctement ?

Question 2)

Je vais créer une fonction qui calcule la taille moyenne de la population à la génération n.

Ma fonction prend deux paramètres :

- n : la génération pour laquelle on veut calculer la taille moyenne
- nb_simulations : le nombre de simulations à effectuer

J'utilise pour calculer les résultats la même fonction que dans la question précédente. Cependant, dans la mesure où je fais un très grand nombre de simulations (ici 1000), je dois recréer une fonction en la modifiant un peu. En effet, cette fonction affiche à chaque tour de boucle le nombre d'individus. Je dois supprimer cette partie de la fonction car sinon je vais afficher plusieurs milliers de fois le nombre d'individus.

J'ai choisi là aussi dans cette fonction d'afficher les résultats pour chaque génération jusqu'au rang n afin d'avoir plus de contexte et de compréhension de mes valeurs.

J'obtiens les résultats suivants :

```
Génération 0 : taille moyenne = 0.94
Génération 1 : taille moyenne = 1.036
Génération 2 : taille moyenne = 1.03
Génération 3 : taille moyenne = 1.01
Génération 4 : taille moyenne = 1.044
Génération 5 : taille moyenne = 0.98
```

```
Génération 0 : taille moyenne = 1.014
Génération 1 : taille moyenne = 0.988
Génération 2 : taille moyenne = 1.004
Génération 3 : taille moyenne = 1.01
Génération 4 : taille moyenne = 1.014
Génération 5 : taille moyenne = 1.1
```

Ainsi pour plusieurs simulations j'ai des résultats similaires. De plus si on calcule l'espérance mathématique de la loi suivie par X : $E\left[X_{i,n}\right] = 2 \cdot P\left(X_{i,n} = 2\right) + 0 \cdot P\left(X_{i,n} = 0\right)$ = 1

Or lors de mes simulations j'obtiens toujours des résultats proches de 1, la fonction fonctionne donc correctement.

Question 3)

Il faut maintenant calculer la valeur de $\pi_n = P[Z_n = 0]$

Je vais donc créer une fonction pour calculer la probabilité d'extinction a pour une génération n choisie. Cette fonction fera appel à la fonction de la question 1, dans un souci comme dans la question 2 de ne pas afficher 1000 fois les valeurs du nombre d'individus je vais utiliser la fonction recrée comme à la question 2. J'ai choisi la aussi de mettre une boucle pour afficher toutes les probabilités jusqu'au rang n voulu afin de voir l'évolution

Pour cela, je réalise un grand nombre de simulations et pour chacune de ses simulations je regarde si la génération s'est éteinte au rang n (=0 individu), si elle s'est éteinte cela renvoie 1 sinon 0 et à la fin je fais la moyenne de toutes les simulations pour obtenir la proba et j'obtiens les résultats suivants :

```
Probabilité d'extinction à la génération 1 : 0.474
Probabilité d'extinction à la génération 2 : 0.575
Probabilité d'extinction à la génération 3 : 0.715
Probabilité d'extinction à la génération 4 : 0.74
Probabilité d'extinction à la génération 5 : 0.769
```

J'ai choisi de commencer d'afficher les probas à la génération 1 car à la génération 0 la proba est de 0 car l'on commence forcément avec 1 individu, cela n'a donc pas de sens de l'afficher.

Je remarque que les probas augmentent ceci est cohérent car bien que l'espérance mathématique soit de 1, il existe encore une probabilité non nulle et importante que l'espèce s'éteigne à chaque génération donc plus la génération est élevée plus la probabilité d'extinction sera élevée. Ma fonction fonctionne donc correctement.

Question 4)

On doit maintenant calculer $P_{Extinction}$.

Pour cela je crée une fonction qui simule les générations tant que la population est supérieure à 0 donc que l'espèce ne s'est pas éteinte. Si elle finit par s'éteindre je renvoie 1 sinon 0. Je fais ensuite la moyenne des valeurs renvoyées par ma simulation et j'obtiens toujours le même résultat : 1

Ce résultat est cohérent avec ce que j'ai trouvé à la question précédente ou la proba d'extinction tendait vers 1 plus le nombre de générations générées était élevé. Je peux aussi confirmer ça par le fait que l'on sait qu'un processus de Galton-Watson à une extinction inévitable pour $E[X] \leq 1$

Ainsi ma fonction fonctionne correctement.

Question 5)

On doit maintenant calculer le nombre moyen de générations avant d'observer l'extinction.

Pour cela je crée donc une fonction qui va simuler un grand nombre de fois (j'ai choisi 1000 car pour n = 10000 le temps de simulation devenait beaucoup trop long, ce temps est parfois déjà assez long pour n=1000) et va regarder à chaque simulation à quelle génération elle s'est éteinte, il va ensuite faire la moyenne des résultats de toutes les simulations pour obtenir le nombre moyen de générations avant extinction.

Ainsi j'obtiens des résultats différents en fonction des simulations, cela peut s'expliquer par la nature très aléatoire du processus, en effet certaines simulations peut s'éteindre très rapidement tandis que d'autres peuvent mettre un grand nombre de générations avant de s'éteindre.

```
Proba extinction :  1
Nombre moyen avant extinction :  8.578
>
```

```
Proba extinction :  1
Nombre moyen avant extinction :  7.98
>
```

A noter que la simulation peut parfois prendre beaucoup de temps en fonction du résultat, ne pas hésiter à stopper le code et à le regénérer si cela prend trop de temps.

Ainsi, ma fonction fonctionne correctement.

Exercice 2

Question 6)

Dans cette question, nous devons simuler comme dans l'exercice 1 l'état de la population à un rang n mais cette fois ci les paramètres de la loi sont différents.

Ainsi je reprends ma fonction de la q1 Exo1 et modifie les paramètres de la loi binomiale pour avoir B(3,p). Ici nous avons 0<p<1, j'ai choisi de ne pas définir mon p et de laisser le choix à l'utilisateur pour définir ce p lorsqu'il appelle la fonction. Ainsi, j'effectue une vérification au début pour vérifier que l'utilisateur choisit bien un p compris entre 0 et 1 sinon la fonction renvoie un message d'erreur. J'ai choisi d'afficher l'état pour chaque génération jusqu'au rang n afin de vérifier que ma fonction fonctionne correctement et qu'il n'y a pas d'erreur.

Ainsi, j'obtiens les résultats suivants :

```
Génération 0 B(3, 0.5 ) : 1
Génération B(3, 0.5 ) 1 : 3
Génération B(3, 0.5 ) 2 : 2
Génération B(3, 0.5 ) 3 : 2
Génération B(3, 0.5 ) 4 : 2
Génération B(3, 0.5 ) 5 : 3
```

```
Génération 0 B(3, 0.5 ) : 1
Génération B(3, 0.5 ) 1 : 2
Génération B(3, 0.5 ) 2 : 3
Génération B(3, 0.5 ) 3 : 4
Génération B(3, 0.5 ) 4 : 5
Génération B(3, 0.5 ) 5 : 7
```

J'affiche le p choisi par l'utilisateur (ici 0,5) pour mieux se repérer si l'on fait plusieurs simulations avec des p différents.

J'obtiens des résultats cohérents car ici un individu peut avoir jusqu'à trois enfants (car l'on fait 3 itérations de la loi binomiale, ainsi il peut avoir 0,1,2,3 enfant(s)).

Ma fonction fonctionne donc correctement.

Question 7)

Même principe que pour la question 2 de l'exo 1, je dois re utiliser la même fonction que la question précédente mais dans la mesure ou je fais beaucoup de simulations pour ne pas afficher plusieurs milliers de valeurs, je recrée la fonction sans afficher le nombre d'individus à chaque génération.

Le code est ensuite sur la base de celui de la question 2 de l'exo 1 mais j'ai modifié la loi pour que ce soit la loi Binomiale B(3,p) ici le p est là encore choisi par l'utilisateur lorsqu'il appelle la fonction. J'ai choisi la aussi dans cette fonction d'afficher les résultats pour chaque génération jusqu'au rang n afin d'avoir plus de contexte et de compréhension de mes valeurs.

Voici les résultats que j'ai obtenu :

```
Génération 0 : taille moyenne = 2.267
Génération 1 : taille moyenne = 1.574
Génération 2 : taille moyenne = 2.191
Génération 3 : taille moyenne = 3.585
Génération 4 : taille moyenne = 5.21
Génération 5 : taille moyenne = 7.373
```

```
Génération 0 : taille moyenne = 0.766
Génération 1 : taille moyenne = 0.832
Génération 2 : taille moyenne = 0.786
Génération 3 : taille moyenne = 0.664
Génération 4 : taille moyenne = 0.68
Génération 5 : taille moyenne = 0.499
```

Les premiers résultats correspondent à p=0,5 tandis que les deuxièmes à p=0,3.

Ainsi je remarque des différences entre les choix de p.

Ma fonction renvoie bien les valeurs des tailles moyennes pour chaque génération, elle fonctionne correctement.

Question 8)

Je vais cette fois ci re utiliser la fonction de la question précédente en changeant mon rang n, je vais choisir n = 10 pour avoir une plus grande plage de donnée :

Pour n = 10 et p= 0.1 j'obtiens :

```
Génération 0 : taille moyenne = 0.08
Génération 1 : taille moyenne = 0.289
Génération 2 : taille moyenne = 0.094
Génération 3 : taille moyenne = 0.026
Génération 4 : taille moyenne = 0.01
Génération 5 : taille moyenne = 0.004
Génération 6 : taille moyenne = 0
Génération 7 : taille moyenne = 0
Génération 8 : taille moyenne = 0
Génération 9 : taille moyenne = 0
```

Ainsi je vérifie bien que pour $p < \frac{1}{3}$ il y a bien extinction de la réaction en chaîne.

Pour n = 10 et p = 0.8 j'obtiens :

```
Génération 0 : taille moyenne = 5.768
Génération 1 : taille moyenne = 2.438
Génération 2 : taille moyenne = 5.781
Génération 3 : taille moyenne = 13.984
Génération 4 : taille moyenne = 32.763
Génération 5 : taille moyenne = 79.655
Génération 6 : taille moyenne = 192.015
Génération 7 : taille moyenne = 455.776
Génération 8 : taille moyenne = 1111.48
Génération 9 : taille moyenne = 2684.28
Génération 10 : taille moyenne = 6431.003
```

Ainsi je vérifie bien que pour $p > \frac{1}{3}$ la réaction en chaîne n'est pas contrôlée

Enfin, pour n = 10 et $p = \frac{1}{3}$ j'obtiens :

```
Génération 0 : taille moyenne = 1.014
Génération 1 : taille moyenne = 0.974
Génération 2 : taille moyenne = 0.969
Génération 3 : taille moyenne = 0.969
Génération 4 : taille moyenne = 1.062
Génération 5 : taille moyenne = 0.995
Génération 6 : taille moyenne = 1.039
Génération 7 : taille moyenne = 0.934
Génération 8 : taille moyenne = 0.992
Génération 9 : taille moyenne = 1.078
Génération 10 : taille moyenne = 1.061
```

Ainsi, la réaction semble maitrisée, elle ne s'éteindra pas et ne sera pas hors de contrôle.

Question 9)

Dans cette question, on doit calculer la valeur de $\pi_n = P[Z_n = 0]$ pour différentes valeurs.

Je reprends ainsi la même fonction que pour la question 4 de l'exercice 1 mais je dois le modifier un peu. Je choisis le nombre de simulations (ici 1000 pour éviter des temps de calculs trop long), pour chaque simulation je regarde si il y a eu une extinction, si oui j'ajoute 1 au nombre d'extinctions et je fais le rapport du nombre d'extinctions par rapport au nombre de simulations pour obtenir la proba (même principe que question 4).

Nous avons beaucoup de valeurs à tester et j'aurais pu les écrire une par une mais cela aurait été un peu long, j'ai remarqué que ces valeurs allaient de 0.1 à 0.5 par pas de 0.05 c'est pourquoi j'ai pu les stocker dans un vecteur (pas besoin d'utiliser une liste car tous les éléments sont du même type). Je défini ensuite le nombre de générations que je veux afficher (ici j'ai choisi 5). Enfin pour présenter les résultats de manière lisible et pratique pour l'utilisateur j'ai choisi de les afficher avec une matrice. Cette dernière affiche le p choisi ainsi que la génération et les probas. Elle a un nombre de lignes égal au nombre de p testé et un nombre de colonnes égal au nombre de générations souhaitées. Enfin je peux choisir la précision obtenue, ici je choisis 3 chiffres après la virgule.

Ainsi j'obtiens les résultats suivants :

```
         n = 0 n = 1 n = 2 n = 3 n = 4 n = 5
p = 0.1  0.910 0.744 0.927 0.978 0.995 0.997
p = 0.15 0.825 0.642 0.833 0.915 0.966 0.987
p = 0.2  0.719 0.519 0.736 0.835 0.917 0.948
p = 0.25 0.630 0.405 0.631 0.724 0.819 0.870
p = 0.3  0.534 0.346 0.515 0.625 0.675 0.754
p = 0.35 0.403 0.257 0.402 0.526 0.572 0.623
p = 0.4  0.329 0.213 0.319 0.364 0.438 0.456
p = 0.45 0.250 0.162 0.249 0.289 0.301 0.304
p = 0.5  0.176 0.128 0.186 0.174 0.211 0.217
```

Ces résultats sont cohérents car l'on remarque que plus p est petit plus la proba d'extinction est élevée et inversement plus p est grand plus elle est faible. On remarque bien la bascule vers p=1/3 où la proba passe de >1/2 à <1/2.

Ainsi tous les résultats sont cohérents, la fonction fonctionne correctement.

Question 10)

On doit maintenant calculer $P_{Extinction}$ pour chacune des valeurs de p

Pour cela je reprends la même fonction que dans la question 4 que je vais modifier un peu. Je choisis le nombre pour lequel une simulation est considérée comme non éteinte (ici j'ai pris 10 car sinon les calculs étaient trop longs), je choisis ensuite le nombre de simulations (j'ai ici aussi pris 10). Les nombres que j'ai choisi sont en effet un peu petit, des nombres plus grands augmenteraient la précision des réponses cependant mon ordinateur met trop de temps à calculer pour des nombres plus grands.

Je regarde donc pour chaque simulation le nombre d'extinctions et je calcule la proba en fonction du nombre de simulations et du nombre d'extinctions. Les valeurs étant les mêmes que dans la question précédente j'utilise la même méthode et les stocks dans un vecteur.

Voici les résultats que j'obtiens :

```
p = 0.10 : PExtinction : 1.000
p = 0.15 : PExtinction : 1.000
p = 0.20 : PExtinction : 1.000
p = 0.25 : PExtinction : 1.000
p = 0.30 : PExtinction : 1.000
p = 0.35 : PExtinction : 0.500
p = 0.40 : PExtinction : 0.600
p = 0.45 : PExtinction : 0.300
p = 0.50 : PExtinction : 0.100
```

Ainsi, nous remarquons bien que pour p<1/3 $P_{Extinction}$ est bien égal a 1 et pour p>1/3 $P_{Extinction}$ est bien inférieur à 1

Question 11)

On doit maintenant calculer une valeur approchée du nombre moyen de générations avant d'observer l'extinction de la réaction en chaîne. Pour cela je reprends ma fonction de la dernière question de l'exo 1 que je modifie un peu.

J'ajoute les valeurs de p que l'on veut tester dans mon vecteur, ici on ne teste que pour p<1/3 car sinon la fonction serait infinie (en effet, pour p>1/3 comme nous avons pu le voir, l'extinction n'arrive pas avec certitude).

Ainsi, j'obtiens les résultats suivants :

```
p = 0.10 : Nombre moyen de générations avant extinction : 1.36
p = 0.15 : Nombre moyen de générations avant extinction : 1.73
p = 0.20 : Nombre moyen de générations avant extinction : 2.05
p = 0.25 : Nombre moyen de générations avant extinction : 2.84
p = 0.30 : Nombre moyen de générations avant extinction : 4.74
```

Ainsi, on peut voir que plus p est élevé plus le nombre moyen de générations avant extinction est élevé, ce qui est cohérent avec le fait que la proba diminue lorsque p augmente.

Ma fonction fonctionne donc correctement.

Question 12)

Dans cette question, nous devons simuler comme dans l'exercice 1 l'état de la population à un rang n mais cette fois ci les paramètres et la loi sont différents.

Ainsi je reprends ma fonction de la q1 Exo1 et modifie les paramètres de la loi binomiale pour avoir ma loi de poisson, j'utilise pour cela 'Rpois'. J'ai choisi de ne pas définir mon λ et de laisser le choix à l'utilisateur pour définir ce λ lorsqu'il appelle la fonction. J'ai choisi d'afficher l'état pour chaque génération jusqu'au rang n afin de vérifier que ma fonction fonctionne correctement et qu'il n'y a pas d'erreur.

Voici les résultats obtenus :

```
Génération 0 : 1
Génération 1 : 3
Génération 2 : 9
Génération 3 : 14
Génération 4 : 20
Génération 5 : 27
```

```
Génération 0 : 1
Génération 1 : 1
Génération 2 : 1
Génération 3 : 0
```

Les premiers sont pour λ = 1.5 et les seconds sont pour λ = 0.5

Mes résultats semblent cohérents d'après les indices de la question 14. Ma fonction fonctionne donc correctement.

Question 13)

Même principe que pour la question 2 de l'exo 1, je dois re utiliser la même fonction que la question précédente mais dans la mesure où je fais beaucoup de simulation pour ne pas afficher plusieurs milliers de valeurs, je recrée la fonction sans afficher le nombre d'individus à chaque génération.

Le code est ensuite sur la base de celui de la question 2 de l'exo 1 mais j'ai modifié la loi pour que ce soit la loi de poisson ici le λ est là encore choisis par l'utilisateur lorsqu'il appelle la fonction. J'ai choisi là aussi dans cette fonction d'afficher les résultats pour chaque génération jusqu'au rang n afin d'avoir plus de contexte et de compréhension de mes valeurs.

Voici les résultats que j'ai obtenu :

```
Génération 1 : taille moyenne = 1.47
Génération 2 : taille moyenne = 2.262
Génération 3 : taille moyenne = 3.357
Génération 4 : taille moyenne = 4.913
Génération 5 : taille moyenne = 7.684
```

```
Génération 1 : taille moyenne = 0.505
Génération 2 : taille moyenne = 0.215
Génération 3 : taille moyenne = 0.138
Génération 4 : taille moyenne = 0.086
Génération 5 : taille moyenne = 0.024
```

Les premiers sont pour λ = 1.5 et les seconds pour λ 0.5

J'affiche les résultats que à partir de la génération 1 car l'on connait la taille moyenne de la génération 0 qui est toujours la même (1).

Les résultats sont là aussi cohérents, j'observe une augmentation de la taille moyenne pour λ=1.5 et une diminution pour λ=0.5. Ma fonction fonctionne correctement

Question 14)

Je vais cette fois ci re utiliser la fonction de la question précédente en changeant mon rang n, je vais choisir n = 10 pour avoir une plus grande plage de donnée :

Pour n = 10 et λ = 0.5 j'obtiens :

```
Génération 1 : taille moyenne = 0.484
Génération 2 : taille moyenne = 0.255
Génération 3 : taille moyenne = 0.11
Génération 4 : taille moyenne = 0.079
Génération 5 : taille moyenne = 0.026
Génération 6 : taille moyenne = 0.017
Génération 7 : taille moyenne = 0.003
Génération 8 : taille moyenne = 0.008
Génération 9 : taille moyenne = 0
Génération 10 : taille moyenne = 0
```

Ainsi je vérifie bien que pour λ<1 il y a surement extinction de la lignée

Pour n = 10 et λ = 1.5 j'obtiens :

```
Génération 1 : taille moyenne = 2.446
Génération 2 : taille moyenne = 6.471
Génération 3 : taille moyenne = 15.877
Génération 4 : taille moyenne = 39.088
Génération 5 : taille moyenne = 94.552
Génération 6 : taille moyenne = 235.558
Génération 7 : taille moyenne = 583.273
Génération 8 : taille moyenne = 1566.09
Génération 9 : taille moyenne = 3878.899
Génération 10 : taille moyenne = 9227.083
```

Ainsi, je vérifie bien que pour λ >1 la réaction devient non contrôlée

Pour n = 10 et λ = 1 :

```
Génération 1 : taille moyenne = 1.026
Génération 2 : taille moyenne = 1.087
Génération 3 : taille moyenne = 1.034
Génération 4 : taille moyenne = 0.88
Génération 5 : taille moyenne = 0.979
Génération 6 : taille moyenne = 0.931
Génération 7 : taille moyenne = 0.981
Génération 8 : taille moyenne = 0.902
Génération 9 : taille moyenne = 1.079
Génération 10 : taille moyenne = 0.939
```

Ainsi, j'observe que la réaction pour λ = 1 semble maitrisée, elle ne s'éteindra pas et ne sera pas hors de contrôle.

Question 15)

Dans cette question, on doit calculer la valeur de $\pi_n = P[Z_n = 0]$ pour différentes valeurs.

Je reprends ainsi la même fonction que pour la question 4 de l'exercice 1 mais je dois le modifier un peu. Je choisis le nombre de simulations (ici 1000 pour éviter des temps de calculs trop long), pour chaque simulation je regarde s'il y a eu une extinctions, si oui j'ajoute 1 au nombre d'extinctions et je fais le rapport du nombre d'extinction par rapport au nombre de simulations pour obtenir la proba (même principe que question 4 et 9).

Je stock dans un vecteur toutes les valeurs de λ que l'on souhaite tester. Je définis ensuite le nombre de générations que je veux afficher (ici j'ai choisi 5). Enfin pour présenter les résultats de manière lisible et pratique pour l'utilisateur j'ai choisi de les afficher avec une matrice.

Ainsi j'obtiens les résultats suivants :

	n = 0	n = 1	n = 2	n = 3	n = 4	n = 5
λ = 0.25	0.947	0.779	0.944	0.993	0.996	0.999
λ = 0.5	0.832	0.623	0.826	0.904	0.954	0.982
λ = 0.75	0.679	0.463	0.655	0.808	0.855	0.903
λ = 1	0.525	0.391	0.515	0.621	0.686	0.725
λ = 1.5	0.306	0.232	0.316	0.343	0.361	0.394
λ = 2	0.179	0.142	0.160	0.189	0.172	0.197
λ = 5	0.008	0.001	0.009	0.003	0.005	0.007

Ces résultats sont cohérents car l'on remarque que plus λ est petit plus la proba d'extinction est élevée et inversement plus p est grand plus elle est faible. On remarque bien la bascule vers λ = 1 ou la proba passe de >1/2 à <1/2.

Ainsi tous les résultats sont cohérents, la fonction fonctionne correctement.

Question 16)

On doit maintenant calculer $P_{Extinction}$ pour chacune des valeurs de λ

Pour cela je reprends la même fonction que dans la question 4 et 10 que je vais modifier un peu. Je choisis le nombre pour lequel une simulation est considérée comme non éteinte (ici j'ai pris 10 car sinon les calculs étaient trop long), je choisis ensuite le nombre de simulation (j'ai ici aussi pris 10). Les nombres que j'ai choisis sont en effet un peu petits, des nombres plus grands augmenteraient la précision des réponses cependant mon ordinateur met trop de temps à calculer pour des nombres plus grands.

Je regarde donc pour chaque simulation le nombre d'extinctions et je calcule la proba en fonction du nombre de simulations et du nombre d'extinctions. Les valeurs étant les mêmes que dans la question précédente j'utilise la même méthode et les stocks dans un vecteur.

Voici les résultats que j'obtiens :

```
Résultats :
λ = 0.25 : PExtinction = 1.000
λ = 0.50 : PExtinction = 1.000
λ = 0.75 : PExtinction = 1.000
λ = 1.00 : PExtinction = 0.700
λ = 1.50 : PExtinction = 0.400
λ = 2.00 : PExtinction = 0.100
λ = 5.00 : PExtinction = 0.000
```

Ainsi on remarque que plus lambda est grand, plus $P_{Extinction}$ diminue. On remarque aussi que la probabilité est de 1 pour λ < 1 et elle tombe même à 0 pour λ = 5.

Ainsi ces résultats sont cohérents avec ce que l'on a pu observer précédemment, ma fonction fonctionne correctement.

Question 17)

On doit maintenant calculer une valeur approchée du nombre moyen de générations avant d'observer l'extinction de la réaction en chaîne. Pour cela je reprends ma fonction de la dernière question de l'exo 1 que je modifie un peu.

J'ajoute les valeurs de λ que l'on veut tester dans mon vecteur, ici on ne teste que pour λ < 1 car sinon la fonction serait infinie (en effet, pour λ >1 comme nous avons pu le voir, l'extinction n'arrive pas avec certitude).

Ainsi, j'obtiens les résultats suivants :

```
λ = 0.1 : Nombre moyen de générations avant extinction : 1.09
λ = 0.2 : Nombre moyen de générations avant extinction : 1.23
λ = 0.3 : Nombre moyen de générations avant extinction : 1.36
λ = 0.4 : Nombre moyen de générations avant extinction : 1.52
λ = 0.5 : Nombre moyen de générations avant extinction : 1.72
λ = 0.6 : Nombre moyen de générations avant extinction : 1.97
λ = 0.7 : Nombre moyen de générations avant extinction : 2.40
λ = 0.8 : Nombre moyen de générations avant extinction : 2.89
λ = 0.9 : Nombre moyen de générations avant extinction : 3.88
```

Ainsi, on peut voir que plus λ est élevé plus le nombre moyen de générations avant extinction est élevé, ce qui est cohérent avec le fait que la proba diminue lorsque p augmente.

Ma fonction fonctionne donc correctement.