

# Introduction aux logiciels pour les statistiques

Examen IS3 / 2020-2021 / Tous documents autorisés

*Le but de l'examen est de vous permettre de montrer vos acquis. Le sujet est volontairement long. Il est donc conseillé de traiter en premier lieu les questions sur lesquelles vous vous sentez le plus à l'aise. Les langages R et Python sont les seuls autorisés pour répondre au sujet. Vous répondrez à chaque question en utilisant au moins une ligne de code et en formulant au moins une phrase de réponse. Recopier une valeur lue visuellement dans la table de données ou l'environnement pour répondre à une question ne rapporte aucun point. Une valeur de la table ne pourra être utilisée qu'à condition de l'avoir trouvée par du code. Pour répondre à une question, vous pouvez admettre le résultat de la question précédente à condition de le mentionner. À la fin de l'examen, vous enregistrerez votre travail dans un fichier zip à votre nom. Vous l'envoyez ensuite à Sofiane.Maazi@polytech-lille.fr. La clarté des commentaires et la qualité du code constitueront un élément important dans l'appréciation de votre travail. Pour chaque question, vous êtes invité(e) à faire figurer toute trace de recherche même infructueuse. Le barème est donné à titre indicatif.*



## Descriptif des tables

<b>Table « cards »</b>	<b>Variables :</b> <ul style="list-style-type: none"> <li>● <b>ID</b> : identifiant unique de la carte</li> <li>● <b>Name</b> : nom de la carte</li> <li>● <b>Type</b> : type de la carte</li> <li>● <b>Level</b> : niveau de la carte</li> <li>● <b>Race</b> : genre de la carte</li> <li>● <b>Attribute</b> : attribut de la carte</li> <li>● <b>ATK</b> : points d'attaque de la carte</li> <li>● <b>DEF</b> : points de défense de la carte</li> </ul>
<b>Autres tables :</b>	<b>Variable :</b> <ul style="list-style-type: none"> <li>● <b>ID</b> : identifiant unique de la carte</li> </ul>

# Partie 1 : Le deck des joueurs

1. La table *cards* contient l'ensemble des cartes du jeu de duel de monstres.
  - (a) La carte *Sand Gambler* est-elle une carte monstre ou piège ?
  - (b) Quel est le nom de la carte dont l'identifiant est 5958 ?
2. Chaque joueur a constitué une pile de cartes appellée « deck ». Les tables *deckYugi* et *deckPegasus* contiennent le deck de chaque joueur. La  $i^{\text{ème}}$  ligne coïncide avec la carte à la  $i^{\text{ème}}$  position dans le deck. La première ligne correspond conséquemment à la carte au sommet de la pile. La dernière ligne correspond à la dernière carte de la pile.
  - (a) Combien de cartes possède le deck de chaque joueur ?
  - (b) Fusionnez *deckPegasus* et *cards* selon la variable *ID* en une table *deckdescriptifPegasus*. Créez une table *deckdescriptifYugi* sur le même principe.
  - (c) La variable *ID* est-elle ordonnée de la même façon dans les tables *cards* et *deckdescriptifPegasus* ( respectivement *deckdescriptifYugi* ) ?
  - (d) En déduire si l'ordre d'apparition des cartes dans le deck de chaque joueur est respecté dans *deckdescriptifPegasus* ( respectivement *deckdescriptifYugi* ).
3. (a) Le deck de Yugi contient-il plus de cartes magie que le deck de Pegasus ?
  - (b) Quel deck possède le plus de cartes monstres ?
  - (c) Peut-on affirmer que 42% des cartes de Yugi sont des cartes monstres ?
4. Quel est le nom des cartes au moins en double exemplaire dans le deck de Pegasus ?
5. Quel est le nom des cartes pièges en commun entre les decks des 2 joueurs ?
6. Un monstre est détruit si ses points de défense sont strictement inférieurs aux points d'attaque du monstre assaillant. L'un des joueurs possède la carte *Toon Mermaid*. Combien de monstres peut-il détruire en théorie dans le deck l'adversaire grâce à cette carte ?
7. Pour quel(s) attribut(s) le nombre moyen de points d'attaque des monstres de Yugi est-il le plus élevé ?
8. La moitié des monstres de Yûgi ont-ils au moins 1000 points de défense ?
9. Proposez une représentation graphique de la variable *DEF* du deck d'un des joueurs.
10. La variable *ATK* du deck de ce même joueur pourrait-elle suivre une loi normale de paramètre  $\mu = 5$  et  $\sigma = 17$  ou une loi de Weibull de paramètres  $\alpha = 2$  et  $\beta = 5$  ?
11. Proposez une analyse descriptive de la variable *ATK* du deck d'un des joueurs.



## Partie 2 : Le duel de monstres

1. On modélise un joueur par une liste en R ou un dictionnaire en Python. Son 1<sup>er</sup> élément nommé *LP* est le nombre de points de vie du joueur. Le 2<sup>nd</sup> élément nommé *main* contient l'identifiant des cartes dans la main du joueur. Le 3<sup>ème</sup> élément est nommé *deck* du joueur et contient l'identifiant des cartes dans le deck du joueur. En début de partie, chaque joueur dispose de 8000 points de vie , de l'intégralité de son deck et n'a aucune carte en main. Les 3 éléments sont des vecteurs de R (ou des listes si vous programmez en Python). Modélez chaque joueur.
2. (a) Écrivez une fonction *Pioche (n, joueur)* qui enlève les n premières cartes du deck et les ajoute à la main du joueur. Elle renverra le nouvel état du joueur.  
(b) Écrivez une fonction *ChoixCarte (ID, joueur)* qui enlève l'une des cartes d'identifiant *ID* du deck du joueur et l'ajoute à la main du joueur. Elle renverra le nouvel état du joueur.  
(c) Écrivez une fonction *JouerCarte (ID, joueur)* qui supprime l'une des cartes d'identifiant *ID* de la main du joueur. Elle renverra le nouvel état du joueur.  
(d) Écrivez une fonction *PointVie (i, joueur)* qui ajoute un nombre de points de vie ( positif ou négatif) *i* aux points de vie du joueur. Elle renverra le nouvel état du joueur.  
(e) En début de partie, les joueurs piochent 5 cartes chacun. Pegasus commence ensuite à jouer. Il pioche une carte de son deck. Il joue la carte *Toon World* et perd 1000 points de vie. En déduire à l'aide des fonctions précédentes, le nouvel état du joueur.
3. (a) Combien de cartes devrait piocher Pegasus dans son deck avant de piocher une carte monstre dont le nom contient le mot « Toon » ?  
(b) Écrivez une fonction *MelangeDeck (joueur)* qui mélange le deck du joueur. Elle renverra le nouvel état du joueur.  
(c) Pegasus joue la carte *Toon Table of Content*. Elle lui permet de regarder son deck et d'ajouter un monstre Toon de son deck à sa main. Pegasus choisi la carte *Toon Ancient Gear Golem*. Il mélange ensuite son deck. En déduire le nouvel état du joueur.
4. (a) Dans les questions suivantes, on ne s'intéresse plus à l'état des joueurs. On s'intéresse désormais au plateau de jeu. On le représente par une matrice vide avec 4 lignes et 5 colonnes. Créez une variable *Terrain* modélisant le plateau de jeu.  
(b) Une carte est dite posée sur le terrain quand son identifiant est inscrit dans *Terrain*. Écrivez une fonction *PoserCarte(ID, i, j)* qui affecte la valeur *ID* à la i<sup>ème</sup> ligne et la j<sup>ème</sup> colonne de *Terrain*. Les cartes magie et piège de Pegasus (respectivement Yugi) seront posées sur la ligne 1 (respectivement ligne 4) . Les cartes monstres de Pegasus (respectivement Yugi) seront posées sur la ligne 2 (respectivement ligne 3) .  
(c) Pegasus (respectivement Yugi) pose à un moment sur le terrain les cartes d'identifiants 3559, 5332, 1249 , 5280, 4336 et 588 (respectivement d'identifiants 3081, 3839 et 3). Posez ces cartes sur le terrain.



## Partie 3 : La théorie des jeux

1. Pegasus ordonne à Toon Eyes Dragon d'attaquer le Magicien Sombre. Yugi active alors la carte piège chapeaux magiques. Trois chapeaux géants apparaissent sur le terrain. Le Magicien Sombre est caché sous l'un d'entre-eux. Pegasus doit désormais choisir d'attaquer l'un des chapeaux. Si le Magicien sombre est sous le chapeau ciblé alors le Magicien Sombre est détruit. Notons A, B et C les 3 chapeaux. La probabilité que Pegasus choisisse le chapeau A est de  $1/8$ . La probabilité que Pegasus choisisse le chapeau B est de  $1/6$ . Simulez le choix de chapeau opéré par Pegasus.



2. Yugi active la carte *Sand Gamble*. Il est écrit sur la carte « Lancez une pièce 3 fois. Si vous obtenez 3 fois Face, détruisez tous les monstres sur le Terrain adverse. Si vous obtenez 3 fois Pile, détruisez tous les monstres sur votre Terrain. ». Écrivez une fonction *Gamble* qui simule le lancer de pièce décrit par la carte *SandGamble* et détruit sur le terrain tous les monstres du joueur malchanceux. Vous préciserez les paramètres de votre fonction et la testerez sur l'état du plateau de jeu de la Partie 2.

3. Pegasus est le créateur du jeu de cartes de duel de monstres. Il a créé une carte unique *Loi paranormale*. Il est écrit dessus « Générez une valeur M d'une variable X suivant une loi normale  $\mu = 7$  et  $\sigma = 3$ . Soit N et P les 1<sup>er</sup> et 2<sup>ème</sup> chiffres après la virgule de M. Soit Q le nombre formé par les 2 chiffres après la virgule de M. Sommez les chiffres du nombre R = Q - (N+P). Si le résultat obtenu est supérieur à  $\sigma$ , vous remportez le duel. » Effectuez plusieurs simulations. Expliquez pourquoi cette carte est interdite lors des duels de monstres.

4. On s'intéresse désormais à quelques règles du duel de monstres.

- (a) Jusqu'ici les joueurs possédaient un deck, des cartes en main et des cartes posées sur le plateau. En réalité, ils possèdent chacun également un « cimetière ». C'est un paquet constitué par les cartes qui ont été détruites sur le terrain ou défaussées par le joueur. Supposons que le cimetière de Yugi contient les cartes dont les ID vont de 3642 à 3663. Créez le data.frame *cimetiereYugi* représentant le cimetière de Yugi.
- (b) La puissance du deck d'un joueur est donnée par la fonction ci-dessous. Implémentez la fonction puissance puis testez la sur le deck de Yugi pour  $\alpha = 10$  et  $\beta = 3$ . Donnez un inconvénient de cette fonction.

$$puissance(x, \alpha, \beta) = \beta \sum_{i=1}^{n-\alpha} x_i^{\beta-1} + \frac{\alpha}{2} \sum_{i=1}^{\beta} \exp(x_i)$$

où  $x$  est la variable numérique ID du deck du joueur,  $n$  est le nombre de cartes dans le deck du joueur,  $\alpha$  et  $\beta$  sont des entiers.

- (c) Pegasus active la carte *Doppelganger le Mimétique*. Il est écrit « Ciblez une carte dans le cimetière de votre adversaire. Si c'est une carte monstre posez la face retournée. Sinon posez la face cachée ». Proposez une fonction Doppel qui pose une carte de *cimetiereYugi* sur le terrain. Proposez également une représentation originale du terrain pour connaître désormais l'état de la carte posée ( face cachée ou retournée). Vous envisagerez toutes les modifications sur les éléments du jeu qu'implique un retour d'une carte depuis le cimetière sur le terrain.



# Barème

## Partie 1 ( 22 points)

1. Numéros des questions notées sur 0.5 points : 1)a) - 1)b) - 2)a) - 2)b) - 2)c) - 2)d)
2. Numéros des questions notées sur 1 point : 3)a) - 3)b) - 3)c) - 7)
3. Numéros des questions notées sur 2 points : 4) - 5) - 9)
4. Numéros des questions notées sur 3 points : 6) - 10) - 11)

## Partie 2 ( 18 points)

1. Numéros des questions notées sur 1 point : 1) - 2)a) - 2)b) - 2)c) - 2)d) - 3)b) - 4)a - 4)b)
2. Numéros des questions notées sur 2 points : 3)c) - 4)c)
3. Numéros des questions notées sur 3 points : 2)e) - 3)a)

## Partie 3 ( 16 points)

1. Numéro des questions notées sur 2 points : 1) - 4)a) - 4)b)
2. Numéro des questions notées sur 3 points : 2) - 3)
3. Numéro de la questions notée sur 4 points : 4)c)

## Source des images [ en ligne ]

- yugioh.fandom.com
- otk-expert.fr
- www.deviantart.com
- dlpng.com