

<https://github.com/exo7math/python2-exo7/blob/master/bigdata/bigdata-1.pdf>

### Activité 3 (La formule des tanks).

Plus d'info :

[https://fr.wikipedia.org/wiki/Probl%C3%A8me\\_du\\_char\\_d%27assaut\\_allemand](https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_char_d%27assaut_allemand)

Pour aller plus loin :

<https://fr.wikipedia.org/wiki/Capture-marquage-recapture>

#### Objectifs :

déterminer la taille  $N$  d'une série  $1, \dots, N$  en ne connaissant que quelques numéros tirés au hasard. En plein milieu de la seconde guerre mondiale les Allemands produisent un nouveau tank plus performant. Les Alliés s'inquiètent car ils ne savent pas combien de ces nouveaux tanks sont produits. Les services de renseignements estiment la production à 1500 tanks par mois.

Que disent les mathématiques ?

Les Alliés ont intercepté 4 tanks produits le même mois et qui portent les numéros : 143, 77, 198 et 32.

Combien de tanks ont été produit ce mois ?

#### Modélisation :

Sachant que les tanks sont numérotés de 1 à  $N$  chaque mois, à quelle valeur peut être estimée la production mensuelle  $N$ , connaissant un échantillon de  $k$  numéros  $[n_1, n_2, \dots, n_k]$  ?

1. La formule des tanks. On note  $m$  le maximum des éléments de l'échantillon. On note  $k$  la taille de l'échantillon. Alors la formule des tanks estime :

$$N = m + \frac{m}{k} - 1$$

 Programme cette formule en une fonction `formule_tanks(echantillon)` qui renvoie cette estimation de  $N$ .

Quelle est ton estimation pour le nombre de tanks ?

2. Le double de la moyenne. On peut essayer d'autres estimations. Par exemple on peut estimer  $N$  comme le double de la moyenne de l'échantillon.

 Programme une fonction `double_moyenne(echantillon)` qui renvoie cette nouvelle estimation. Compare avec la formule des tanks.

 Pour tester l'efficacité de la formule des tanks **on va faire le cheminement inverse** : on fixe un entier  $N$ , on choisit au hasard un échantillon de  $k$  éléments et on regarde si nos formules permettent de bien approcher  $N$ .

 3. Tirage sans remise. Programme une fonction `tirage_sans_remise(N,k)` qui renvoie une liste de  $k$  entiers différents compris entre 1 et  $N$  (inclus).



### Correction des 3 premières questions

Dans le fichier texte suivant se trouve le script que vous pouvez copier/coller dans votre IDE. Ce script contient les trois fonctions des exos 1, 2 et 3.

<https://www.ipa-troulet.fr/cours/attachments/article/542/correction.txt>

 4. Erreurs. Programme une fonction `erreurs(N,k,nb_tirages=1000)` qui calcule l'erreur moyenne commise par nos formules. Pour cela :

- Effectue un tirage sans remise de  $k$  entiers plus petits que  $N$ .
- Calcule la valeur  $N1$  obtenue à partir de cet échantillon par la **formule des tanks**.
- Calcule l'erreur commise  $e = |N - N1|$ .
- Réapplique ce principe pour les 1000 tirages puis calcule la moyenne des 1000 erreurs commises

Fais le même travail avec l'autre formule et renvoie les deux erreurs moyennes.

 5. Pour 20 entiers plus petits que 1000 ( $k = 20$  et  $N = 1000$ ) quelle est la meilleure formule et à quelle erreur peut-on s'attendre ?

—

 *À la fin de la guerre, les registres Allemands ont été récupérés et indiquaient une production de 245 chars mensuels ! Cette formule est aussi utilisée pour estimer la production d'un produit (par exemple d'un téléphone) à partir des numéros de série.*



### Correction des parties 4 et 5

Je vous laisse les faire;)

#### **Remarque :**

*Cette activité est dans un document pdf portant le nom **bigdata**, sauriez-vous m'en donner la raison ?*

—