

# Test Data Analyst - Analyse B-VZXR

---

## Objectif

On a retrouvé les logs de robots qu'il faudra analyser. Ces robots s'appellent des B-VZXR et ont des fonctionnements bien particuliers. Vous devrez, dans cet exercice, comprendre les comportements des différents types de B-VZXR via l'analyse de leurs logs.

## B-VZXR

À chaque robot correspond un espace quadrillé de hauteur et largeur fixe.

Ces espaces sont donc un ensemble de cases. Chaque case est représentée par ses coordonnées (x, y) où x et y sont des entiers positifs. La case tout en bas à gauche a pour coordonnées (0, 0).

Un robot commence chacun de ses parcours sur la même case.

À tout moment un robot se définit par :

- sa position : coordonnées de la case sur laquelle se trouve le robot
- sa direction : un des quatre points cardinaux, orientation de la tête du robot, vers laquelle il peut avancer

Nous savons également que le B-VZXR ne sait lire qu'un générer qu'un seul type d'instructions. Ces instructions sont composées de deux informations :

1. une rotation, soit `right`, soit `left`
2. une valeur entière positive

Lorsqu'il génère une instruction, le B-VZXR tourne sa tête vers sa droite si l'instruction est `right`, ou vers sa gauche si l'instruction est `left`. Ensuite il avance du nombre de cases indiquées par la deuxième partie de l'instruction. S'il est en train d'avancer et qu'il rencontre un mur, il arrête d'avancer et génère une nouvelle instruction. On parle alors de `wall_hit`.

## Exemple parcours

On considère un B-VZXR qui vit dans un espace de 6 longueur et de 4 de largeur. Qui commence sur la case (0, 0), la tête en direction du nord.

On considère que le B-VZXR génère la liste d'instructions suivantes :

1. `right, 3`
2. `left, 2`
3. `left, 1`
4. `left, 0`
5. `left, 3`
6. `right, 1`
7. `right, 10`
8. `right, 2`

Dans ce cas là, le robot va lire la première instruction `right, 3` et va donc se tourner vers la droite et avancer de trois cases. Il sera alors dans la case (3, 0). Après avoir exécuté la seconde instruction `left, 2`, il sera sur la case (3, 2)... La suite de son voyage est lisible sur l'image ci-dessous.

 B-VZXR accross the universe

On notera que B-VZXR lit l'instruction `right, 10`, il rencontre un mur et va donc rester sur la case en face du mur. Il lira ensuite la prochaine instruction. À la fin de son périple, il va se retrouver sur la case (0, 3).

## Logs

Les B-VZXR aiment bien communiquer. Ils transmettent à chacun de leurs voyages les messages suivants :

### Logs agrégés

- **robot\_id** (chaîne de caractères) : son identifiant
- **experiment\_id** (chaîne de caractères) : l'identifiant de son parcours (un robot effectue plusieurs parcours)
- **distance** (entier) : la distance parcourue par le robot
- **nb\_{cardinal}** (entier) : le nombre de fois que le robot a pris la direction cardinale en question
- **nb\_{cardinal}\_wall\_hit** (entier) : le nombre de fois que le robot a tapé dans le mur correspondant à la direction cardinale indiquée

### Logs de position

- **experiment\_id** (chaîne de caractères) : l'identifiant de son parcours (un robot effectue plusieurs parcours)
- **current\_instruction** (entier|chaîne de caractères) : le nombre d'instructions lues, **final** représente la dernière position du robot
- **x** (entier) : la position du robot après avoir lu le nombre d'instructions à **current\_instruction**
- **y** (entier) : la position du robot après avoir lu le nombre d'instructions à **current\_instruction**

Chaque robot effectue plusieurs parcours dans son espace prédéfini.

## Exemple de logs

Le B-VZXR de l'exemple aura donc les logs suivants :

<b>robot_id</b>	<b>experiment_id</b>	<b>distance</b>					
B-VZXR_0000	B-VZXR_0000_1	17					
<b>nb_north</b>	<b>nb_east</b>	<b>nb_south</b>	<b>nb_west</b>	<b>nb_north_wall_hit</b>	<b>nb_east_wall_hit</b>	<b>nb_south_wall_hit</b>	<b>nb_west_wall_hit</b>
2	2	2	2	0	0	0	1

et

<b>experiment_id</b>	<b>current_instruction</b>	<b>x</b>	<b>y</b>
B-VZXR_0000_1	2	3	2
B-VZXR_0000_1	5	5	2
B-VZXR_0000_1	10	NaN	NaN
B-VZXR_0000_1	final	0	3

## Indications sur le choix de l'espace du robot et la génération d'instructions

Il faut caractériser chaque robot selon les critères suivants :

- La taille de l'espace dans lequel il évolue ((6, 4) dans l'exemple)
- Sa position initiale ((0, 0) dans l'exemple)
- La façon dont le robot génère ses instructions (qui sont des comportements aléatoires) :
  - la distance qu'il parcourt à chaque instruction (l'exemple ne suffit pas à répondre à cette question)
- Un robot est cassé et ne peut pas effectuer un type d'action. Il faut identifier le robot.

## Rendu

### Dataset

Nous vous fournissons, avec cet énoncé, deux fichiers qui représentent les logs de 10 robots différents. Chaque robot a généré 10 000 parcours.

## Fichiers

- un fichier `aggregated_logs.csv` de 100 000 lignes de logs agrégés
- un fichier `path_logs.csv` de 600 000 lignes de logs de positions

## Analyse attendue

Votre objectif consiste à identifier et montrer les différences comportementales des robots. Il vous faudra, pour chaque robot, définir les valeurs suivantes :

- `height` : la hauteur de l'espace quadrillé dans lequel il évolue
- `width` : la largeur de l'espace quadrillé dans lequel il évolue
- `pace` : une estimation de la distance parcourue à chaque déplacement
- `init_x` : la position de départ du robot
- `init_y` : la position de départ du robot

Afin de les définir vous pourrez :

1. Calculer, pour chaque robot, les descriptifs des colonnes (min, max, moyenne, médiane...). Le descriptif des colonnes des deux fichiers est attendu.
2. Représenter graphiquement les distances parcourues par chaque robot
3. Représenter graphiquement les informations cardinales de chaque robot
4. Représenter graphiquement la distribution des positions pour chaque robot et chaque `current_instruction` (ou dans des cas précis qui vous semblent judicieux)
5. Toute représentation graphique qui vous semblera pertinente

## Outils recommandés

Nous ne vous demandons pas d'utiliser un outil spécifique. Nous recommandons de le faire en Python mais vous pouvez utiliser d'autres outils (Excel, R...)

## Format

Vous devrez donc rendre :

1. un fichier contenant votre analyse (LibreOffice, Word, PDF...)
2. le code qui vous a permis de générer vos visualisations (Python, Excel, R...) (Si vous choisissez d'utiliser Excel, il est inutile de nous renvoyer le fichier avec tout le dataset, un échantillon suffira)

## Critères d'évaluation

- Respect rigoureux des consignes ci-dessus
- Clarté du rendu
- Pertinence des visuels choisis
- Qualité du raisonnement