**1. results :**

**C'est la section principale du fichier qui contient les résultats de la course.**

**2. c2\_race\_id : id**

**3. duration :**

**La durée totale de la course, ici définie à 2000, qui correspond à la distance totale que chaque rameur doit parcourir (2000 mètres).**

**4. ergrace\_version :**

**Cela indique la version du logiciel ou du système utilisé pour enregistrer les données. Par exemple : "02.01.17".**

**5. participants :**

**C'est une liste d'objets, chaque objet représentant un rameur qui a participé à la course. Chaque rameur a ses propres données spécifiques détaillées. Voici une explication de chaque champ pour un participant typique :**

**Pour chaque participant (participants) :**

1. **affiliation** : Ce champ pourrait contenir l'affiliation du rameur, par exemple son équipe, son club ou son université. Dans ton exemple, ce champ est vide.
2. **avg\_pace** : C’est la **vitesse moyenne** du rameur, généralement exprimée en minutes et secondes par 500 mètres, c’est le temps moyen qu’il faut au rameur pour parcourir 500 mètres.
3. **calories** : Le nombre total de **calories brûlées** par le rameur pendant la course. Cela peut être utilisé pour estimer l'effort physique fourni.
4. **distance** : La distance totale parcourue par le rameur pendant la course. Dans ton cas, cela est toujours de 2000 mètres, car chaque rameur doit couvrir cette distance.
5. **lane** : C’est le numéro de **voie** du rameur, comme dans une course de natation ou d'athlétisme, chaque rameur a une voie assignée. Par exemple, "lane": 4 signifie que ce rameur était dans la voie numéro 4.
6. **logged\_time** : C’est l’**heure à laquelle la course a été enregistrée** pour ce rameur. Par exemple, "22/01/2023 10:59:00" signifie que la course a été enregistrée à 10h59 le 22 janvier 2023.
7. **machine\_type** : Indique le type de machine utilisée pour la course. Ici, "row" indique que c’est un ergomètre de rameur.
8. **participant** : Le **nom du participant**. Par exemple, "Maxime".
9. **place** : La **position** du rameur à la fin de la course. Si "place": 1, cela signifie que ce rameur est arrivé premier dans sa course.
10. **score** : Le **temps total** mis par le rameur pour terminer les 2000 mètres. Par exemple, "6:33.4" signifie que ce rameur a parcouru les 2000 mètres en 6 minutes et 33,4 secondes.
11. **serial\_number** : Le **numéro de série** de la machine utilisée. Cela peut être utile pour identifier quelle machine a été utilisée dans quelle course.

**Pour chaque portion de 500m (splits) :**

Ensuite, pour chaque participant, il y a une liste splits qui contient des informations sur la performance du rameur à chaque portion de 500 mètres. Voici les champs clés pour chaque portion (appelée **split**) :

1. **split\_avg\_pace** : La **vitesse moyenne** sur cette portion de 500 mètres. Par exemple, "1:41.6" signifie que le rameur a mis en moyenne 1 minute et 41,6 secondes pour parcourir ces 500 mètres.
2. **split\_calories** : Le nombre de **calories brûlées** sur cette portion de 500 mètres.
3. **split\_distance** : La distance parcourue pour cette portion, qui est généralement 500 mètres.
4. **split\_stroke\_count** : Le nombre total de **coups de rame** pendant cette portion. Cela permet de savoir combien de coups de rame le rameur a utilisés pour parcourir les 500 mètres.
5. **split\_stroke\_rate** : La **cadence** de rame, c’est-à-dire le nombre de coups de rame par minute pour cette portion. Cela indique l’intensité de l’effort sur cette distance.
6. **split\_time** : Le **temps total** mis pour parcourir cette portion de 500 mètres. Par exemple, "1016" signifie que le rameur a mis 1016 secondes pour parcourir 500 mètres.

# Chemin vers le dossier contenant les fichiers JSON

folder\_path = 'C:/Users/fanti/Music/universite paris-sorbonne/python/rameur/' # Remplacez par le chemin correct vers votre dossier

# Initialisation d'une liste vide pour stocker les données

all\_data = [] # Initialiser ici

# Parcourir tous les fichiers JSON dans le dossier, la condition if file\_name.endswith('.json') vérifie que seuls les fichiers JSON sont sélectionnés.

for file\_name in os.listdir(folder\_path):

if file\_name.endswith('.json'):

file\_path = os.path.join(folder\_path, file\_name)

# Charger le fichier JSON

with open(file\_path, 'r', encoding='utf-8') as file:

data = json.load(file)

# Extraire la liste des participants

participants = data['results']['participants']

# Parcourir les participants et extraire les informations

for participant in participants:

splits = participant['splits']

# Calculer les totaux pour chaque rameur

total\_time\_2000m = participant['score'] # le temps total de score pour parcourir 2000m

total\_cadence\_2000m = participant['spm'] # la cadence moyenne le nombre de coups par minute sur 2000m

total\_strokes\_2000m = sum([split['split\_stroke\_count'] for split in splits]) # le nombre total de coups de rame sur 2000m

# Créer les informations pour chaque rameur, en séparant les portions de 500m en colonnes individuelles

participant\_info = {

'Participant': participant['participant'],

'Temps 2000m': total\_time\_2000m,

'Cadence Moyenne 2000m': total\_cadence\_2000m,

'Coups de Rame 2000m': total\_strokes\_2000m,

'Temps 500m1': splits[0]['split\_time'],

'Cadence 500m1': splits[0]['split\_stroke\_rate'],

'Coups 500m1': splits[0]['split\_stroke\_count'],

'Temps 500m2': splits[1]['split\_time'],

'Cadence 500m2': splits[1]['split\_stroke\_rate'],

'Coups 500m2': splits[1]['split\_stroke\_count'],

'Temps 500m3': splits[2]['split\_time'],

'Cadence 500m3': splits[2]['split\_stroke\_rate'],

'Coups 500m3': splits[2]['split\_stroke\_count'],

'Temps 500m4': splits[3]['split\_time'],

'Cadence 500m4': splits[3]['split\_stroke\_rate'],

'Coups 500m4': splits[3]['split\_stroke\_count']

}

# Ajouter les données du dictionnaire à la liste

all\_data.append(participant\_info)

# Créer un DataFrame avec les informations qui se trouvent dans all\_data

final\_df = pd.DataFrame(all\_data)

# Styliser le DataFrame avec des colonnes bien séparées et des en-têtes en gras

styled\_df = final\_df.style.set\_properties(\*\*{ 'padding': '5px'})\

.set\_table\_styles([{'selector': 'th', 'props': [('font-weight', 'bold')]}])\

.set\_caption("Performances des rameurs sur 2000m avec des portions de 500m")

# Afficher le tableau stylisé

styled\_df