



BIENVENUE SUR LE : SPORTS ANALYTICS CHALLENGE

Afin de tester tes compétences et de prouver que les sciences et le sport n'ont aucun secret pour toi, nous te proposons un jeu de données fourni par la société Opta.

Ton résultat devra être proposé en langage de programmation **Python ou R**, à partir de fichiers Opta F24. Ton code devra être rapide et court. Le jury se basera sur **la performance de ton code** pour classer les finalistes. Temps de calcul sur fichier test : **5 secondes maximum** pour le fichier de la base test sur un ordinateur portable.

Précisions générales :

- Chaque fichier doit être envoyé séparément dans ton espace de participation. Un dossier .zip ne sera pas étudié.
- Le code ne doit comporter que des chemins relatifs et pas de chemin absolu. Tous les fichiers envoyés seront téléchargés et stockés par les organisateurs dans un dossier commun.

INSTRUCTIONS EN PYTHON :

- Version 3.5.
 - Packages autorisés : pas de restriction, toutefois utilise des packages qui peuvent être installés en utilisant *pip*.
 - Tu dois envoyer au moins les 3 fichiers : *main_psgx.py*, *install_psgx.py*, *readme.txt*. Chaque fichier doit être envoyé séparément.
- 1) Le fichier *main_psgx.py* doit contenir une fonction "Result" qui doit prendre en argument un fichier XML identique à celui du fichier test fourni et **rendre un fichier .csv (sans header) nommé res_psgx.csv dans le même dossier, avec les 4 quantités suivantes séparées par une virgule :**
- L'identité du joueur doit être un réel correspondant à son ID.
 - Le prochain événement est un réel qui vaut 1 ou 0.
 - Les prochains y et x sont des réels.



Le test consistera à lancer la fonction "Result(xml_1)" avec

- `xml_1 = lxml.etree.parse(path)`
- `path` = chemin d'accès d'un fichier xml identique à celui présenté dans les ressources sous le nom « Exemple de fichier pour base de test - f24-24-2016-xxxxxx-eventdetails_test_hackaton_2.xml »

2) Le fichier *install_psgx.py* doit contenir la liste des packages à installer. Pour chaque package utilisé bien écrire la commande `os.system()`. Tu es expérimenté ? N'hésite pas à utiliser le package `pipenv` et à l'indiquer.

3) Le fichier *readme.txt* doit contenir :

- Les packages utilisés.
- Une description succincte de chaque sous dossier contenu dans le répertoire soumis.

INSTRUCTIONS EN R :

- Version 3.4
- Packages autorisés: packages disponibles sur CRAN.
- Tu dois envoyer au moins les 3 fichiers : *main_psgx.R*, *install_psgx.R*, *readme.txt*. Chaque fichier doit être envoyé séparément.

1) Le fichier *main_psgx.R* doit contenir une fonction "Result" qui doit prendre en argument un fichier XML identique au fichier test fourni et **rendre un fichier .csv (sans header) nommé res_psgx.csv dans le même dossier, avec les 4 quantités suivantes séparées par une virgule :**

- L'identité du joueur doit être un réel correspondant à son ID qui est un entier.
- Le prochain événement est 1 ou 0 qui est un réel.
- Les prochains y et x sont des réels.

Le test consistera à lancer la fonction "Result(xml_1)" avec

- `xml_1 <- xmlParse(path)`
- `path` = chemin d'accès d'un fichier xml identique à celui présenté dans les ressources sous le nom « Exemple de fichier pour base de test - f24-24-2016-xxxxxx-eventdetails_test_hackaton_2.xml »

2) Packages mode d'emploi : le fichier *install_psgx.R* doit contenir la liste des packages à installer. Pour chaque package utilisé bien écrire la commande `install.packages()`.

3) Le fichier *readme.txt* doit contenir :

- Les packages utilisés.
- Une description succincte de chaque sous dossier contenu dans le répertoire soumis.

Bonne chance !
L'équipe du Sports Analytics Challenge

